

# Wind Turbine Main Bearing Fatigue Life Estimation with Physics-informed Neural Networks

Yigit Anil Yucesan<sup>1</sup> and Felipe A. C. Viana<sup>2</sup>

<sup>1,2</sup> *University of Central Florida, Orlando, FL, 12309, 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 ( $L_{10}$  fatigue life) and the data-driven layers account for the hard to model components (e.g., contribution due to poor greasing conditions).

## 1. INTRODUCTION

As pointed by Hornemann and Crowther (2018), main bearings of onshore wind turbines are subjected to multiple failure modes, among which we can mention wear and micropitting, 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:

- Operating conditions: wind conditions, hazardous weather, machine controls (overload mitigation, yaw misalignment, etc.).

- Environment: wind (extreme cases, turbulence, etc.), ambient temperature, humidity, dust, etc.
- Operator: while some tend to push machines to their limit, others are more zealous.
- 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).

Lubricant condition drastically affects bearing fatigue life. Unfortunately, modeling lubricant performance and degradation is incredibly difficult. Zhu et al. (2015) proposed a methodology for estimating the remaining useful life of lubricant using viscosity and dielectric constant sensor output and integrating these parameters as an observation function by particle filtering technique to predict remaining useful life of the lubricant. Their proposed model was validated by

Yigit Anil Yucesan and Felipe A. C. Viana. 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.

conducted laboratory experiments. Results of the conducted case study show that single observation on dielectric constant sensor gives the best accuracy on life prediction.

This paper proposes overcoming some of limitations in modeling bearing fatigue life by infusing physics into machine learning models. Namely, we propose modeling fatigue life through a recurrent neural network and incorporate a pure data-driven approach to model lubricant degradation. 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 (contribution due to poor greasing conditions).

Recurrent neural networks (RNNs) have been successfully used to model time-series (Connor et al., 1994; Sak et al., 2014; Chauban and Vig, 2015) speech recognition (Graves et al., 2013), and many other applications. Only recently, the scientific community has studied and proposed architectures that leverage formulations based on physics (Yu et al. 2018, Raissi and Karniadakis, 2018, Chen et al. 2018, Ruthotto and Haber, 2018). Differential equations are used to train multi-layer perceptrons and RNNs. Nascimento and Viana (2019) proposed an RNN cell inspired on cumulative damage models (Fatemi and Yang, 1998; Frangopol et al., 2004). These models are often used to describe the irreversible accumulation of damage (progressive distress) throughout the useful life of components or systems. Ultimately, the accumulated damage hits a threshold level that is associated with repair, partial or total replacement, or even worse than that, the retirement, or catastrophic failure of the component or system. The interested reader can also find literature on Gaussian processes (Schober et al., 2014, Raissi et al., 2018).

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.

## 2. PHYSICS-INFORMED MACHINE LEARNING

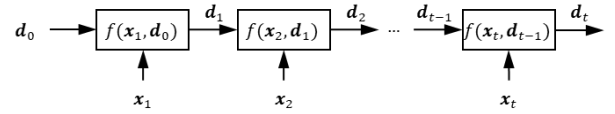
### 2.1. Recurrent Neural Networks and Cumulative Damage Models

RNNs (Goodfellow et al., 2016) transform a vector of hidden states,  $\mathbf{d}$ , in the following fashion:

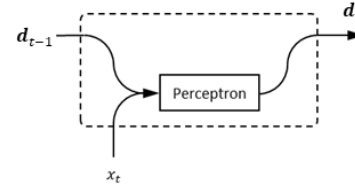
$$\mathbf{d}_t = f(\mathbf{d}_{t-1}, \mathbf{x}_t), \quad (1)$$

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

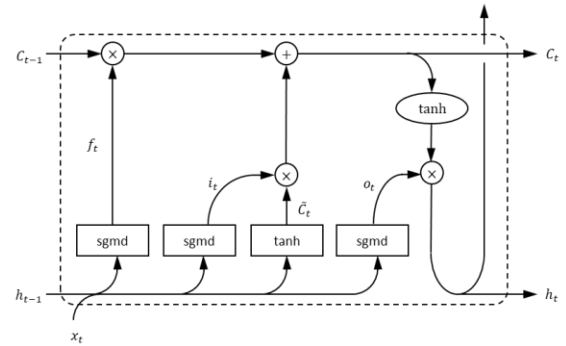
As illustrated in Figure 1-(a), the RNN cells repeatedly apply the transformations to the states. These states can be observed all the time or only at particular time stamps. Figure 1-(b) shows the simplest RNN cell, where a perceptron with a sigmoid activation function maps the inputs at time  $t$  and states at time  $t - 1$  into the states at time  $t$ . Architecture of a single RNN cell can be tailored for desired problem. For example, over time, the scientific community working on data-driven applications have proposed the LSTM (Hochreiter and Schmidhuber, 1997), illustrated in Figure 1-(c). The cell was designed to (a) improve the predictions of the neural network, and (b) mitigate the vanishing gradient problem (Goodfellow et al., 2016).



(a) Recurrent cells of RNN.



(b) Simple RNN cell.



(c) Long short-term memory (LSTM) cell.

Figure 1. Examples of recurrent neural networks (RNN) cells. In the LSTM cell, the squares are perceptrons with pre-defined activation functions; the oval shape is just the **tanh** activation.

Cumulative damage models represent damage at time  $t$  as 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  $t$ .

We use the repeating cell proposed by Nascimento and Viana (2019) to model cumulative damage through RNNs. As illustrated in Figure 2, “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. If a purely physics-based approach is used, “MODEL” is related to the physics of failure (which is highly application dependent). As it will be discussed in next section, “MODEL” is developed to be a hybrid model, where some parts are physics-based while others are data-driven.

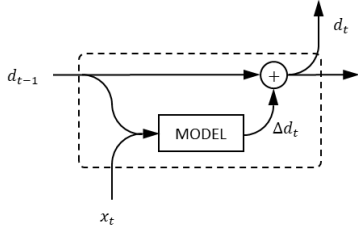


Figure 2. Cumulative damage RNN cell.

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

Bearing fatigue life is parametrized 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 spherical roller bearings catalogue (SKF contributors, 2007), the fatigue life is calculated by

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

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),  $a_1$  is the life adjustment factor for reliability, see Table 1,  $a_{SKF}$  is the SKF life modification factor, see Figure 3,  $C$  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).

The adjustment factor  $a_{SKF}$  depends on the lubrication condition in terms of viscosity (through the viscosity ratio,  $\kappa$ ) and particulate contamination (through the contamination factor,  $\eta_c$ ), the equivalent dynamic bearing load  $P$ , and the fatigue load limit  $P_u$ .  $\kappa$  is expressed as

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

where  $\nu$  actual operating viscosity of the lubricant ( $\text{mm}^2/\text{s}$ ), and  $\nu_1$  rated viscosity, depending on the bearing mean diameter and rotational speed, ( $\text{mm}^2/\text{s}$ ).

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

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

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, the subscript  $t$  indicates the time step and  $n$  is the number of cycles per time step.

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

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

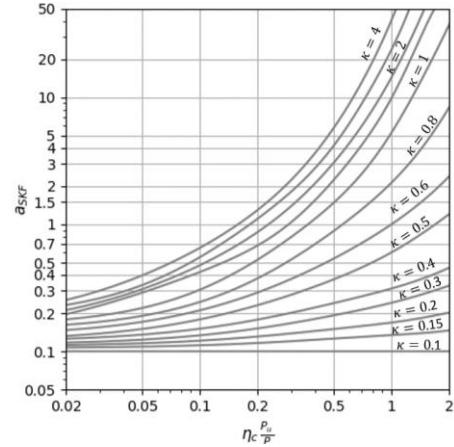
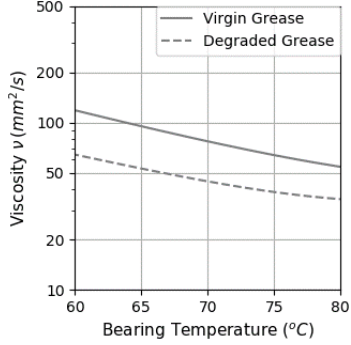


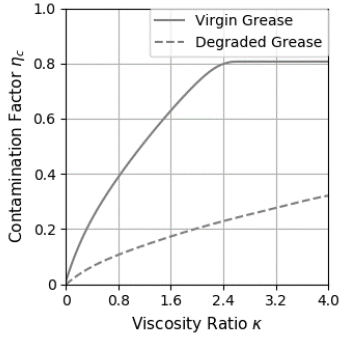
Figure 3.  $a_{SKF}$  life adjustment factor (SKF contributors, 2007).

We used data available in the SKF catalogues to determine the virgin grease curves and then we arbitrarily chose the degraded grease curves. 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 (as per SKF catalogue), while

degraded lubricant would present very severe contamination (as per SKF catalogue). Figure 4 illustrates the variation of grease properties for the virgin and degraded greases.



(a) Grease viscosity vs. bearing temperature



(b) Contamination factor vs. viscosity ratio

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

It is challenging to build a purely physics-based model for bearing fatigue life, since grease degradation is extremely complex. There are attempts to build physics-based models for grease life, but it is not clear how they relate to field conditions (see Appendix A for one example of such models). Here we propose a cumulative damage model for bearing fatigue life that is a hybrid of physics and machine learning.

This way, a recurrent neural network (RNN) can be built such that:

$$\mathbf{d}_t = \mathbf{d}_{t-1} + \Delta \mathbf{d}_t, \quad (7)$$

where:

- $\mathbf{d}_t = [d_t^{BRG} \quad d_t^{GRS}]$ ,

- $d_t^{BRG}$  is the bearing cumulative damage, which damage increment  $\Delta d_t^{BRG}$  is based on published lifing curves for bearing fatigue, and
- $d_t^{GRS}$  is the grease cumulative damage, which damage increment  $\Delta d_t^{GRS}$  is modeled through a multilayer perceptron model.

With that, we propose the repeating RNN cell illustrated in Figure 5 to model the bearing and grease cumulative damage. This RNN cell takes wind speed ( $WS_t$ ) and the bearing temperature ( $T_t$ ) as input variables. The cell will be recurrently used, as in Figure 1-(a), updating both the grease and bearing damages from previous time step ( $\mathbf{d}_{t-1}^{GRS}$  and  $\mathbf{d}_{t-1}^{BRG}$ , respectively). While  $WS_t$  is mapped to equivalent dynamic bearing load ( $P_t$ ) (see Figure 6),  $T_t$  and cumulative grease damage from previous time step ( $\mathbf{d}_{t-1}^{GRS}$ ) are used to calculate grease damage parameters  $\kappa_t$  and  $\eta_{c_t}$  (as in Figure 4). Combined with  $P_t$ , these parameters are incorporated to evaluate inverse life adjustment factor  $1/a_{SKF}$  (see Figure 3), which is then multiplied with non-adjusted bearing fatigue damage increment ( $n_t/N_t$ ) for bearing fatigue damage increment ( $\Delta d_t^{BRG}$ ) calculation (Eq. 3-6). The data-driven portion of the hybrid model is given by prediction of grease damage increment ( $\Delta d_t^{GRS}$ ) via multilayer perceptron<sup>1</sup> (MLP) using  $P_t$ ,  $T_t$ , and  $\mathbf{d}_{t-1}^{GRS}$  as inputs. The MLP architecture is further discussed in Section 3.4. The MLP calculates increments to the grease damage that are then added to damage values from previous time step. In this scheme, while we maintain a physics portion with bearing fatigue accumulation, we also compensate the missing physics knowledge within the grease model with the help of neural networks. The training of this RNN aims at calibrating the MLP using grease damage observations and let it learn the damage accumulation on grease.

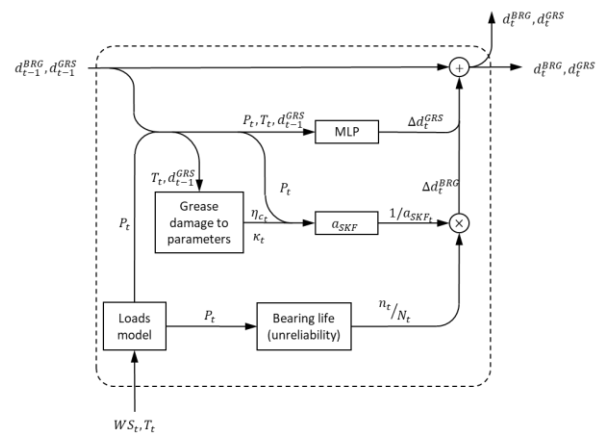


Figure 5. Physics-informed neural network framework for main bearing fatigue and grease degradation.

<sup>1</sup> The LSTM cell uses single layer perceptrons with pre-defined activation functions, shown as squares in Figure 1-(c). Here, our model uses a multilayer perceptron.

### 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 case study.

Table 2. Wind turbine and main bearing specifications

Main bearing	
Designation	SKF 230/600 CAW33
Basic dynamic load rating $C$	6000 kN
Fatigue load limit $P_u$	750 kN
Mass	405 kg
Mean diameter $d_m$	735 mm
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

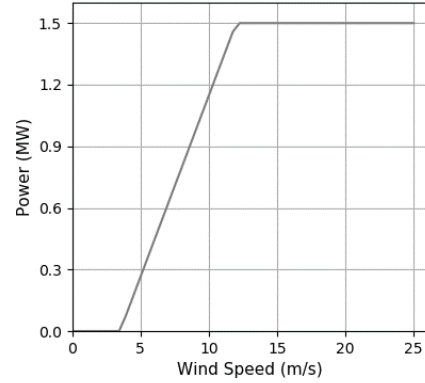
Mapping from wind speed to dynamic bearing loads is maintained using a published National Renewable Energy Laboratory (NREL) report (Sethuraman et al., 2015), which involves a plot that provides dynamic load value for a given wind speed condition for the same type of main bearing we used in our case study, mounted on a 1.5MW wind turbine. Rotational speed output is calculated using the power curve of the wind turbine. Load, power, and rotational speed curves are provided in Figure 6.

#### 3.2. Nominal Wind Speed and Bearing Temperature

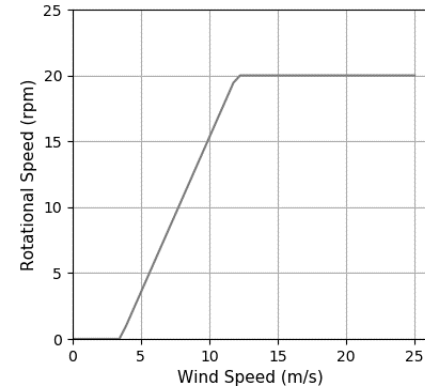
Site-specific data is obtained from a database also provided by NREL (Draxl et al., 2015), which includes environmental data at one hour resolution between 2007 and 2013 for 126,000 different locations throughout the United States. For the present case study, we arbitrarily chose Clayton, NM without 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 (upsampling) to achieve the 10 minute resolution. Data is also extended up to 30 years to be used for long term bearing fatigue life predictions. Details of the data augmentation are given in the Appendix B. 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 Appendix C. At the end, the time series that we consider

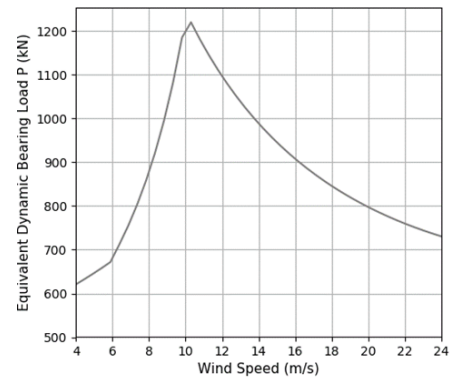
nominal conditions for wind speed and bearing temperature are shown in Figure 7. In order to generate synthetic set of wind turbines we divided 7 years of data into segments of 6 months which yields to 14 different data sets that we treat as 14 different turbines. We partitioned these machines into 10 training and 4 validation turbines.



(a) Power curve



(b) Rotational speed



(c) Equivalent dynamic bearing load

Figure 6. Wind speed mapping for case study turbine.

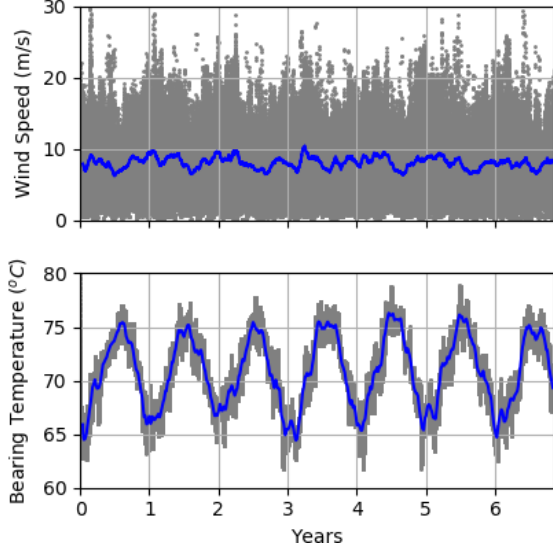


Figure 7. Time series for wind speed and main bearing temperature. Data is represented in gray and trend is plotted in blue (monthly moving average to highlight any seasonality).

### 3.3. Grease samples

In essence, the bearing fatigue model needs information about the viscosity and contamination of grease over time. In real life, one way to obtain these grease parameters is through periodic sampling and laboratory analysis. With the process repeated continuously, the parameters used in bearing fatigue estimation could be updated, which would allow for accurate lifing of the component.

Here, we create synthetic grease samples using the model described in Appendix A. 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 Figure 8):

$$d^{\eta_c} = d^{\kappa} = 1/(L^{GRS})^2 \quad (8)$$

where  $d^{\kappa}$  is damage in terms of viscosity,  $d^{\eta_c}$  is damage in contamination, and  $L^{GRS}$  is the life of grease (see Eq. 14 in Appendix A).

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

$$v = d^{\kappa} (v_{deg} - v_{vir}) + v_{vir}, \quad (9)$$

$$\eta_c = d^{\eta_c} (\eta_{c_{deg}} - \eta_{c_{vir}}) + \eta_{c_{vir}} \quad (10)$$

where  $v$  and  $\eta_c$  are viscosity and contamination factor of the grease respectively.

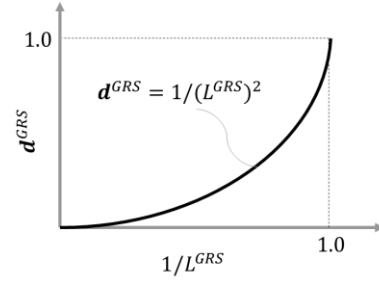


Figure 8. Quadratic relationship between grease life and damage.

Equations (8)-(10) are rather arbitrary and they are only used here as a way to generate synthetic grease sample data. As mentioned before, wind farm operators could obtain data for viscosity and contamination through grease sample analysis. In addition, here we consider the effect of grease degradation on viscosity and contamination as the same. In reality, the effect of damage on different grease parameters might differ from each other. For the sake of the RNN model, the grease damage vector is:

$$\mathbf{d}^{GRS} = \{d^{\kappa}, d^{\eta_c}\} \quad (11)$$

We build our synthetic 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  $d^{\kappa}$  and  $d^{\eta_c}$  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.,  $d^{\kappa} = d^{\eta_c} = 0$  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,  $\mathbf{d}^{GRS}$ , 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,  $\delta \mathbf{d}^{GRS}$ , is a multi-layer perceptron and the bearing damage accumulation is physics-based. Table 3 details the multilayer perceptron used in this work. Dense #0 is a perceptron without activation function that only scales the inputs (weights and bias selected such that inputs are normalized between zero and one). For layers Dense #1 and #5 we used sigmoid activation function, while for layers Dense #2, #3, and #4 we chose *elu* activation

(See Appendix D for details on activation functions). This way, the total parameters number of parameters is 1,263 (out of which 1,251 are trainable and 12 are non-trainable).

Table 3. Grease degradation increment MLP,  $\delta d^{GRS}$ , layer details.

Layer	Output shape	# params	Trainable?
Dense #0	(None, 3)	12	N
Dense #1	(None, 40)	160	Y
Dense #2	(None, 20)	820	Y
Dense #3	(None, 10)	210	Y
Dense #4	(None, 5)	55	Y
Dense #5	(None, 1)	6	Y

We decided to use this architecture to illustrate the ability to fit a neural network with a large number of trainable parameters. No attempt was made to further simplify the multi-layer perceptron. In practical applications, we believe reducing the model is worth pursuing, as it could potentially lead to a more manageable number of trainable parameters without sacrificing accuracy.

The constructed MLP essentially takes three inputs (wind speed, bearing temperature, and current  $d^{GRS}$ ) and provides one output ( $\delta d^{GRS}$ ). However,  $\delta d^{GRS}$  is never observed. Instead, the cumulative damage  $d^{GRS}$  is observed through grease sample laboratory analysis. Here, we used the mean squared error as the loss function while optimizing the trainable parameters of the MLP model within the RNN. 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:

$$Loss = \frac{1}{N_T N_O} \sum_{j=1}^{N_T} \sum_{i=1}^{N_O} (d_{ij}^{GRS} - \hat{d}_{ij}^{GRS})^2 \quad (12)$$

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.

It turns out that optimizing the 1,251 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:

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 \quad (13)$$

where,  $x_i$  are normalized inputs, and  $y$  is the output. The coefficients are initialized using engineering judgement. For example, we assume that  $\delta d^{GRS}$  increases with increasing bearing temperature, therefore the regression coefficient on bearing temperature has to be positive. For illustration purpose, one of the randomly generated plane is plotted against the actual input output relationship in Figure 9. In this illustration, wind speed and bearing temperature are the two inputs of MLP and the grease damage increment  $\delta d^{GRS}$  is the output of the MLP. 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 MLP. Note that the third input variable grease damage is fixed to 0.5 for this plot, in order to make 3D plotting possible.

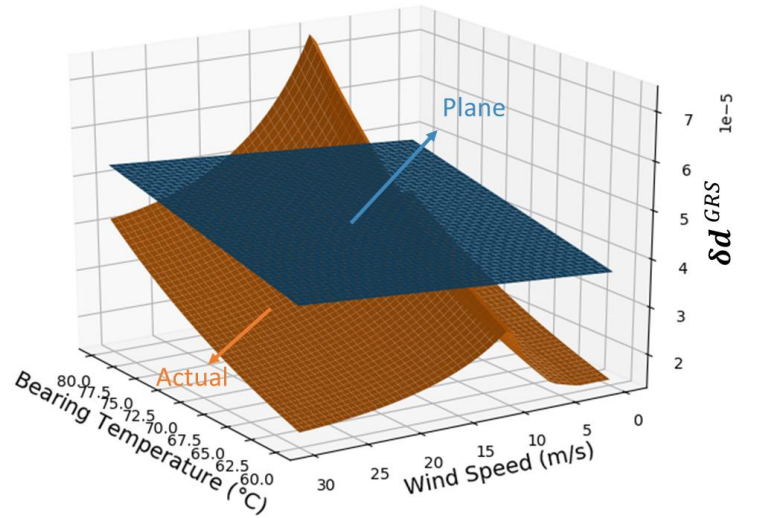


Figure 9. Plane approximation to actual data. Note that in this illustration the third variable,  $d^{GRS}$ , is fixed to 0.5.

We initially train our MLP model with the plane approximation. To achieve that, we used the RMSprop<sup>2</sup> optimizer set with learning rate 0.01 and 500 epochs. We used the mean square error as the loss function. Second stage of the training process is fine tuning the pretrained (i.e., initialized) MLP models inside RNN framework using the masked mean square error given in Eq. 12 as the loss function. Again, we used the RMSprop optimizer, but this time set with learning rate  $5 \times 10^{-4}$  and 50 epochs.

In Section 4. Results and Discussion, we show how the RNN performs when initialized with 10 different randomly

generated  $\alpha_i$  coefficients (all constrained by engineering judgement of how inputs affect the output).

### 3.5. Replication of results

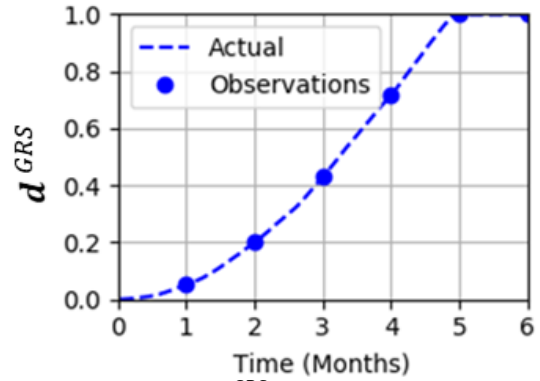
Our implementation is all done in TensorFlow<sup>3</sup> (version 2.0.0-beta1) using the Python application programming interface. In order to replicate the results presented here, the interested reader can download the codes and data. First, install the PINN python package (base package for physics-informed neural networks used in this work) available at Viana et al (2019). Then, clone the “pinn\_wind\_bearing” repository found in Yucesan and Viana (2019b). This repository includes two sets of the code. The basic set of codes contains a script that trains the RNN using a pretrained MLP 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 (1) generates a random plane approximation for MLP training, (2) trains the MLP with randomly generated initial weights, (3) trains the RNN using trained MLP model, and (4) predicts the fatigue damage accumulation of the wind turbine main bearing for 30 years. The data used in this work is also 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).

## 4. RESULTS AND DISCUSSION

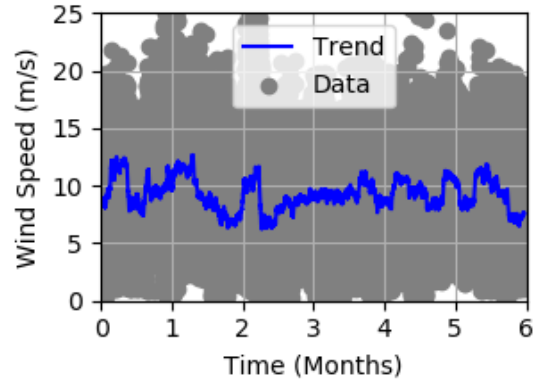
Figure 10 and Figure 11 present the variation of  $d^{GRS}$ , wind speed, and bearing temperature versus time for two wind turbines within the set. This helps visualizing the diversity in our training set. In these figures, blue lines in wind speed and temperature plots show the trend of the data, from which we can observe the seasonality.

As detailed in Section 3, we generated 10 random planes (as exemplified in Figure 9) to initialize the weights and we named these initializations as Case 1-10. After that, we compared the performances of these planes against the actual (but unknown) value of  $\delta d^{GRS}$ , as shown in Figure 13. As we expected, predictions are far away from accurate, since we randomly generated plane coefficients to approximate input-output relationship of MLP. However, in some cases (such as Case 1), trend of the predictions are somewhat aligned with the actual values. Figure 13 provides a good understanding of how the initial approximations may vary from one another. While Case 1 in the Figure 13-(a) is an example of relatively accurate initial approximation, Case 9 shown in Figure 13-(b) is a poor approximation for the input-output relationship. In the presence of engineering intuition or an educated guess about the inputs and output, it is possible to suggest more accurate initial relationship that would enhance the training afterwards. In practice, one would not know how well the

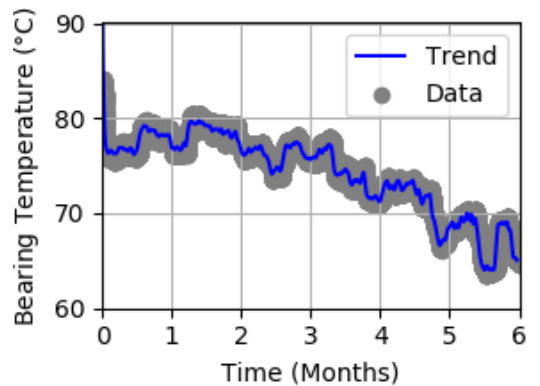
initial weights represent the true behavior of  $\delta d^{GRS}$ . The interesting and challenge case of using RNNs for cumulative damage is that the MLP models a hidden output. In other words, although the cumulative damage  $d^{GRS}$  is observed, the damage increment  $\delta d^{GRS}$  is not. Nevertheless, since we are using synthetic data in this study, we can afford illustrating the performance of the MLP even before training.



(a)  $d^{GRS}$  variation



(b) Wind speed variation



(c) Bearing temperature variation

Figure 10. Turbine 2 time series.

<sup>3</sup> www.tensorflow.org



Figure 12 illustrates the variation of actual  $d^{GRS}$  and observation points (provided to model for training) at the end of each month for six months of duration. We can see that not all  $d^{GRS}$  does not evolve at the same rate across the turbines in the training set (due to difference in the inputs).

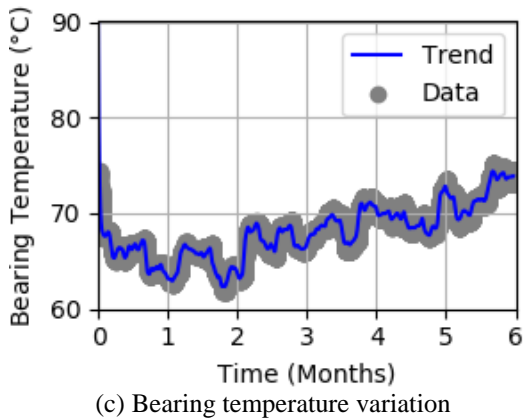
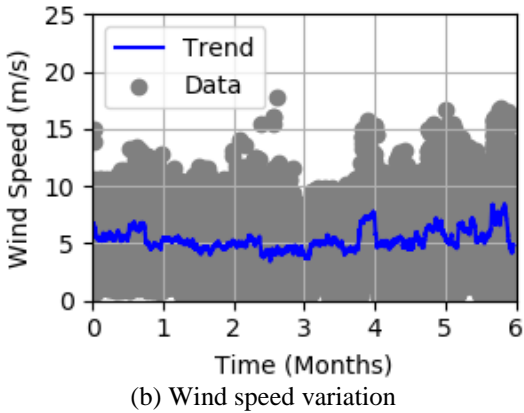
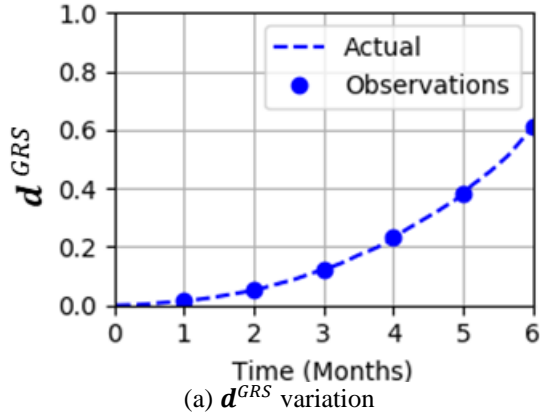


Figure 11. Turbine 9 time series.

After the initialization of the weights, the MLP is integrated into the RNN framework. Figure 14 illustrates the prediction capability of RNN before training, after training, and with validation turbines for two different cases. Unless we are extremely lucky with our random plane generator and have

an excellent approximation of input output relationship, we should expect inaccurate estimations without training. Blue data points in Figure 14 show the inaccuracy of the model before it is trained. After we train our model separately for each cases, we observe predictions are getting better relative to their prior values. The importance of initial weights of the MLP makes a difference here. Using same number of epochs and same optimization settings, while Case 1 almost aligns with 45° line, Case 9 fails to approximate the actual values, as we can observe from red data points. Same trained models are used to predict for validation turbines. Note that not a single data point from validation turbines is used in the training of the model. Black data points in Figure 14 implies that the performance of trained RNN model is almost the same with training and validation turbines. Overall, it is safe to say the model can learn the  $d^{GRS}$  propagation, depending on the initial approximation of the weights of MLP.

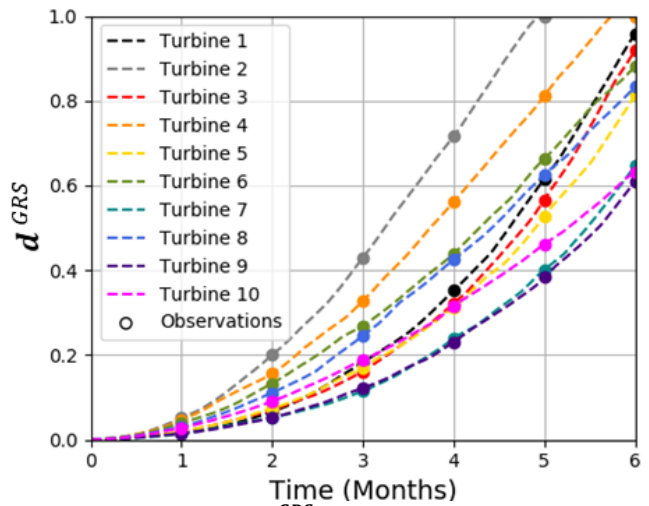


Figure 12. All turbines  $d^{GRS}$  propagation and observations.

In Figure 15, we presented all 10 cases, where we initiated the weights randomly and trained within the RNN framework. As we can see from Figure 15, initiating the weights as a random plane yields to a well-trained model in almost 4 out of 10 cases (Case 1, 2, 3, and 10), fairly well in 3 out of 10 cases (Case 4, 5, and 7), and inaccurate in 3 out of 10 cases (Case 6, 8, and 9). When we look at this figure from a broad perspective, we can conclude that initiating weights using a random plane is an accurate approximation in this study, such that in a fair amount of times it helps the model to capture the behavior of data and trains to smaller errors. Note that errors presented are calculated by subtracting prediction of  $d^{GRS}$  from true value of  $d^{GRS}$  at observation points. It can be also inferred from Figure 15 that the model tends to overestimate the damage in the almost all of the cases. This tendency provides conservatism to the model, which is a preferred type of error in safety assessment applications.

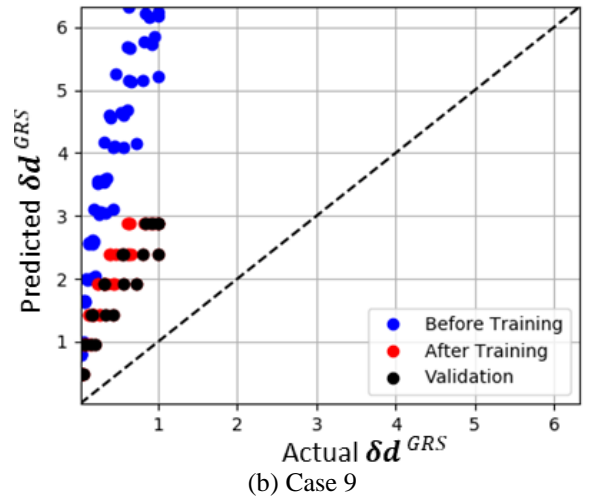
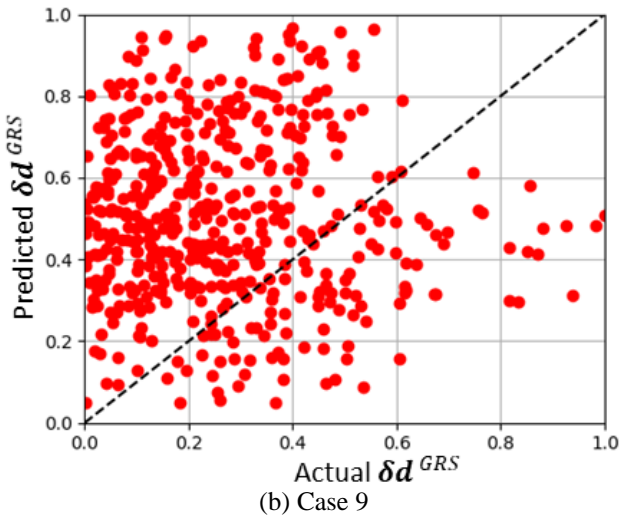
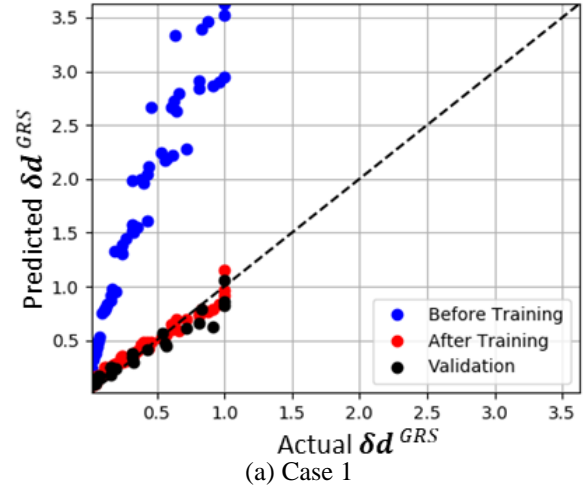
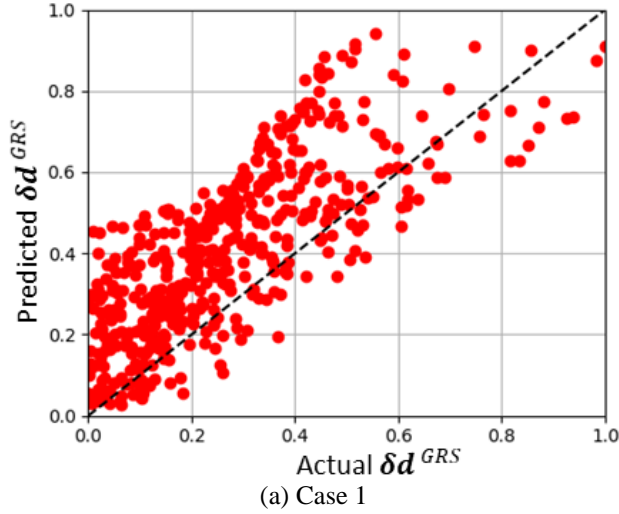


Figure 13. Normalized outputs of randomly generated plane representations against actual output values. In some cases like Case 1 (a), randomly generated approximation provides relatively good results even before training phase. However, as shown in Case 9 (b), some planes are not good representations.

Finally, we implemented our trained models into our larger physics-informed neural network model (Figure 5) to estimate bearing fatigue damage. We chose only three of the best cases we have in the previous simulations (Cases 1, 7, and 10) for convenience. If we compare the prediction results of the model with actual  $d^{GRS}$  variation (as in Figure 17), we can observe the projection of overestimation errors vividly. While Cases 1 and 10 captures the trend of the variation, Case 7 is always off with the prediction. However even Cases 1 and 10 are overestimating the damage value for the most of the time before every 6 months of regreasing cycle.

Figure 14. RNN predictions before and after training, and validation turbines.

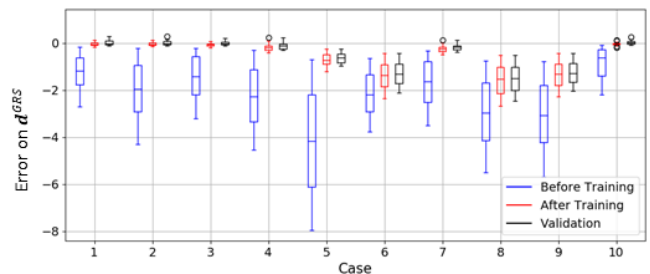


Figure 15. Box plot for before and after training and validation errors.

Figure 16 shows the change in the loss function during the training phase. It is obvious that after only a few epochs, all

cases can converge to a value, which indicates the ability of rapid learning of the model.

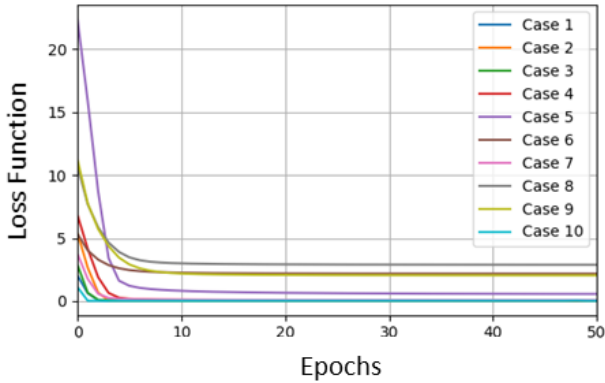


Figure 16. Loss function variation per epoch for all cases.

This error in  $d^{GRS}$  estimation is also reflected on the bearing fatigue damage accumulation. In Figure 18, damage accumulation of different scenarios are illustrated up to failure. Fully degraded and non-damaged (virgin) grease curves represent bearings operating under constant state of grease, failed and pristine states respectively. These two curves form an envelope in the plot, since they are the extreme cases for this machine. Actual curve is where grease degradation is observed and the component is fully regreased every 6 months. Along with these envelopes, dashed curves represent the estimations of our PINN models. As mentioned before, the error in  $d^{GRS}$  estimation causes the model to overestimate bearing fatigue as well. However, our best models (Case 1 and 10) were able to predict the failure only with a few months earlier, which can be described as good estimates relative to ~16 years of total life.

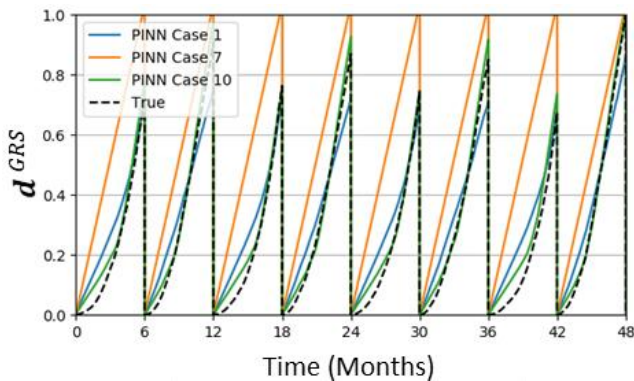


Figure 17.  $d^{GRS}$  propagation for three different cases.

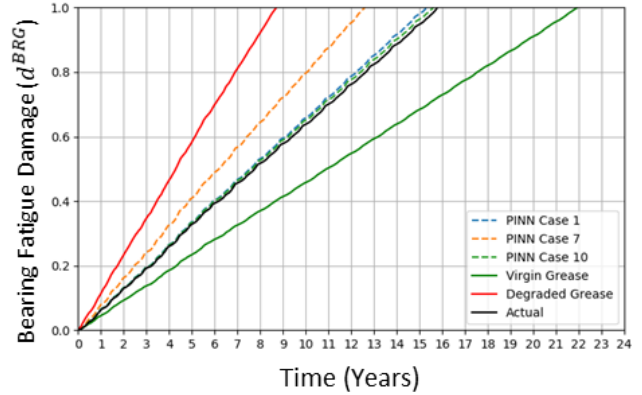


Figure 18. Bearing fatigue damage propagation for three different cases.

## 5. CLOSING REMARKS AND FUTURE WORK

In this contribution we modeled wind turbine main bearing fatigue damage accumulation with a physics-informed machine learning approach. We took the advantage of known physics about how the damage is accumulated throughout each cycle of its life and we leveraged the learning capabilities of deep neural networks to model relatively unknown lubrication effect on the failure mode.

With the help of a numerical study, we learned that:

- initialization of the weights of MLP is crucial: a set of initial weights that is far away from optimum would not lead to accurate prediction,
- the dependency of initial weights can be overcome through engineering-based weight initialization,
- and provided a plausible initial point, artificial neural networks can capture the grease degradation trend with a small error by training only with a few observation points.

Concluding remarks that we inferred from this study are:

- even though prediction errors exist for grease damage increment, we can estimate the fatigue life of the bearing with a decent accuracy,
- the small overestimation in grease damage accumulation yields to a conservative prediction error in bearing fatigue life estimation,
- and we can utilize this hybrid approach on fatigue damage accumulation of wind turbine main bearings for accurate fatigue life prediction.

Building on top of this work, we would like to improve our contribution by:

- improving the design of the MLP to obtain the best prediction capability we can achieve,

- addressing other uncertainties within the model (e.g. loads model, sampling process etc.) and compensating them using deep neural networks,
- and expanding our case study with multiple wind farms located in sites with different environmental conditions, and mature our model to eventually generate a farm level reliability report.

#### ACKNOWLEDGEMENT

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, *Proceedings of IEEE International Conference on Control Applications* (1092–1097), Dubrovnik.
- 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). Anomaly detection in ECG time signals via deep long short-term memory networks. *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*,(1-7). doi:10.1109/DSAA.2015.7344872.
- Chen, T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *31st Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Curran Associates, Inc., (6572—6583).
- Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, 1994, (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 (355–366).
- 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*, Vol. 20, No. 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*, Vol. 6, No. 4,(197-212). doi:10.1002/pse.180.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016) *Deep Learning*, MIT Press, URL: <http://www.deeplearningbook.org>.
- Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, (6645-6649). doi:10.1109/ICASSP.2013.6638947.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, Vol. 9, No. 8, (1735-1780). doi:10.1162/neco.1997.9.8.1735.
- Hornemann, M., & Crowther, A. (2013). Establishing failure modes for bearings in wind turbines. URL: <https://www.windpowerengineering.com/mechanical/bearings/establishing-failure-modes-for-bearings-in-wind-turbines/>. Accessed May 23, 2018.
- Lugt, P. M. (2009). A review on grease lubrication in rolling bearings. *Tribology Transactions*, 52(4), (470-480).
- Nascimento, R. G., & Viana, F. A. C. (2019). Fleet prognosis with physics-informed recurrent neural networks. arXiv preprint, arXiv:1901.05512.
- Raissi, M., & Karniadakis, G. E. (2018). Hidden physics models: machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, Vol. 357, (125 – 141). doi:<https://doi.org/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*, Vol. 40, No. 1., (A172-A198). doi: 10.1137/17M1120762.
- Ruthotto, L., & Haber, E. (2018). Deep neural networks motivated by partial differential equations. *CoRR*, Vol. abs/1804.04272.
- Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Fifteenth annual conference of the international speech communication association*.
- Schaeffler contributors. (2016). Lubrication of Rotor Bearings in Wind Turbines, URL: <http://www.midpointbearing.com/wp-content/uploads/2017/10/FAG-Main-Shaft-Bearing-Lubrication.pdf> [retrieved 2 December 2018].
- Schober, M., Duvenaud, D. K., & Hennig, P. (2014). Probabilistic ODE solvers with Runge-Kutta means. *27th Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Curran Associates, Inc., (739-747).
- Sethuraman, L., Guo, Y., & Sheng, S. (2015). Main bearing dynamics in three-point suspension drivetrains for wind turbines. *American Wind Energy Association Conference & Exhibition*, May 18–21, Orlando, FL.
- SKF contributors. (2007). SKF spherical roller bearings catalogue. URL: [http://www.skf.com/binary/30-148465/6100\\_EN.pdf](http://www.skf.com/binary/30-148465/6100_EN.pdf) [retrieved 5 June 2018]

- Viana, F. A. C., Nascimento, R. G., Yucesan, Y. & Dourado, A. Physics-informed neural networks package, v0.0.2, Zenodo, <https://github.com/PML-UCF/pinn>, doi:10.5281/zenodo.3356877.
- 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), (349-68).
- Yu, Y., Yao, H., & Liu, Y. (2018). Physics-based Learning for Aircraft Dynamics Simulation. *Proceedings of the Annual Conference of the PHM Society*, Vol. 10 (1), September 24-27, Philadelphia, USA. <https://doi.org/10.1234/phmconf.2018.v10i1.513>.
- Yucesan, Y. A. (2019) Wind turbine main bearing fatigue life prediction with PINN. Harvard Dataverse, V1, <https://doi.org/10.7910/DVN/ENNXLZ>.
- Yucesan, Y. A., & Viana, F. A. C. (2019a). Onshore wind turbine main bearing reliability and its implications in fleet management. *AIAA Scitech 2019 Forum*, (p. 1225).
- Yucesan, Y. A. & Viana, F. A. C. (2019b). Python scripts for wind turbine main bearing fatigue life estimation with physics-informed neural networks, v0.0.1, Zenodo, [https://github.com/PML-UCF/pinn\\_wind\\_bearing](https://github.com/PML-UCF/pinn_wind_bearing) [retrieved on Aug/2019], doi:10.5281/zenodo.3355725.
- Zhu, J., Yoon, J. M., He, D., & Bechhoefer, E. (2015). Online particle - contaminated lubrication oil condition monitoring and remaining useful life prediction for wind turbines. *Wind Energy*, 18(6), 1131-1149.

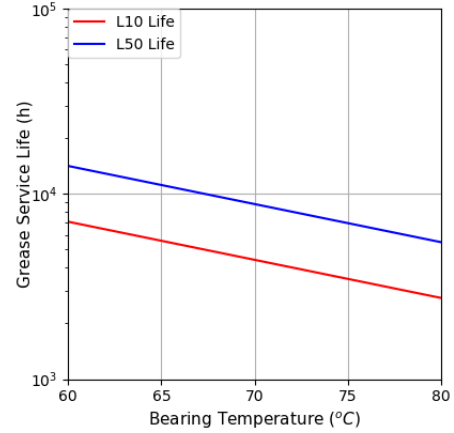
## APPENDIX

### A. GREASE DEGRADATION

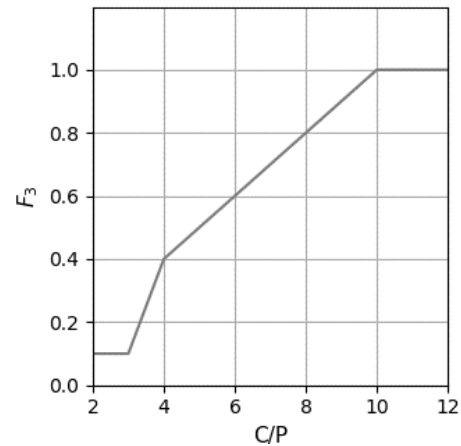
Grease degradation is an extremely complex phenomenon to understand, let alone model. In this paper, we adopted a simplified model that relates grease life with bearing temperature and a number of adjustment factors:

$$L_{nm}^{GRS} = L_{nm}^{GRS*} K_N K_B F_1 F_2 F_3 F_4 F_5 F_6 \quad (14)$$

Figure 19-(a) illustrates how grease service life varies with temperature. Most adjustment factors are given in Table 4.  $F_3$  is a factor that accounts for dynamic load variation and it is shown in Figure 19-(b). 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). As also discussed in the same paper, L50 life of grease can be approximated as the double of L10 life. Similarly to bearing fatigue, we used the Palmgren-Miner's rule.



(a) Nominal grease service life versus bearing temperature



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

Figure 19. Grease life and  $F_3$  adjustment factor.

Table 4. Grease related modification factors and rated viscosity.

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
$\nu_1$	119 mm <sup>2</sup> /s	Rated viscosity

### B. 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 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 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.

### C. 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:

$$= \beta_1 T_{BRG}(t-1) + \beta_2 T_{Nacelle}(t) + \beta_3 N^2(t) + \beta_4 Pwr(t) \quad (15)$$

where:

- $T_{BRG}$  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
- $\beta_i$  are the regression coefficients, see Table 5.

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

Coefficient	Value	Unit
$\beta_1$	0.987	
$\beta_2$	0.0113	
$\beta_3$	0.0115	K s <sup>2</sup> /rad <sup>2</sup>
$\beta_4$	0.0146	K/MW

Most terms in Eq. (15) are easily estimated using the NREL database.  $N$  and  $Pwr$  come from passing the wind speed through the curves shown in Figure 6.  $T_{Nacelle}$  is not available in the NREL database, but we modeled it as a linear function of ambient temperature (which is available in the NREL database):

$$T_{Nacelle}(t) = 0.5 T_{Ambient}(t) + 250 \quad (16)$$

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

### D. ACTIVATION FUNCTIONS

In this study, sigmoid and elu activation functions are used within the MLP layers. These functions are given as follows and Figure 20 illustrates these activation functions.

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (17)$$

$$Elu(x) = \begin{cases} x & \text{if } x > 0 \\ e^x - 1 & \text{if } x < 0 \end{cases} \quad (18)$$

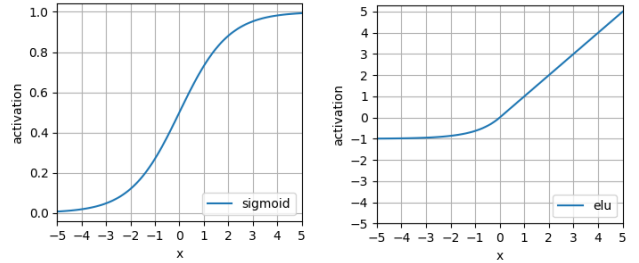


Figure 20. Activation functions.