A hybrid model for main bearing fatigue prognosis based on physics and machine learning

Yigit Anil Yucesan* and Felipe A. C. Viana†
University of Central Florida, Orlando, FL, 32816, USA

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 design lives of the component. 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 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 (e.g., contribution due to poor greasing conditions).

I. Nomenclature

\[ L_{nm} \] = rating life (at 100 − \( n \)) % reliability (in millions of revolutions)
\[ L_{nmh} \] = rating life (at 100 − \( n \)) % reliability (in operating hours)
\[ L_{GRS} \] = grease life (in millions of revolutions)
\[ a_1 \] = life adjustment factor for reliability
\[ a_{SKF} \] = life adjustment factor
\[ C \] = basic dynamic load rating (kN)
\[ P \] = equivalent dynamic bearing load (kN)
\[ N \] = rotational speed (rpm)
\[ \kappa \] = viscosity ratio
\[ \nu \] = viscosity
\[ \nu_1 \] = rated viscosity
\[ \eta_c \] = contamination factor
\[ d^{BRG} \] = cumulative bearing fatigue damage
\[ d^{GRS} \] = cumulative grease damage vector
\[ d^K \] = cumulative grease damage on viscosity
\[ d^{\eta_c} \] = cumulative grease damage on contamination
\[ \delta d^{BRG} \] = incremental bearing fatigue damage
\[ \delta d^{GRS} \] = incremental grease damage vector

II. Introduction

As pointed by Hornemann and Crowther [1] 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:

* Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, yucesan@knights.ucf.edu.
† Assistant Professor, Department of Mechanical and Aerospace Engineering, Sr. AIAA Member, viana@ucf.edu.
• 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. [2] 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 [3]. 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 curtailing. Yucsan and Viana [4] 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. [5] 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 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 layer is used model the relatively well understood physics (L10 fatigue life) and the data-driven layer accounts for the hard to model components (i.e., poor greasing conditions).

Recurrent neural networks have been successfully used to model time-series [6]-[8] speech recognition [9], and many other applications. Only recently, the scientific community has studied and proposed architectures that leverage formulations based on physics [10]-[13] (where differential equations are used to train multi-layer perceptrons and recurrent neural networks). Nascimento and Viana [14], [15] proposed a recurrent neural network cell inspired on cumulative damage models [16], [17]. 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 [18], [19].

The remaining of the paper is organized as follows. Section III gives an overview on physics-informed neural networks and our approach to modeling main bearing fatigue and grease degradation. Section IV describes the case study and the design of the neural network. Section V presents and discusses the numerical results. Finally, section 0 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.

III. Physics-informed machine learning

A. Recurrent neural networks and cumulative damage models

Recurrent neural networks [20] transform a vector of hidden states, $\mathbf{d}$, in the following fashion:

$$d_t = f(d_{t-1}, \mathbf{x}_t),$$

where $t \in [0, ..., T]$ 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 illustrated in Fig. 1-(a), the recurrent neural network cells repeatedly apply the transformations to the states. These states can be observed all the time or only at particular time stamps. Fig. 1-(b) shows the simplest recurrent neural network 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 \( t \). The architecture of a single recurrent neural network cell can be tailored for desired problem. For example, over time, the scientific community working on data-driven applications have proposed the long short-term memory cell [21], illustrated in Fig. 1-(c). The cell was designed to (a) improve the predictions of the neural network, and (b) mitigate the vanishing gradient problem [20].

Cumulative damage models represent damage at time \( t \) as an damage increment \( \Delta d_t \) on top of damage \( d_{t-1} \) at previous time step \( t - 1 \)

\[
d_t = d_{t-1} + \Delta d_t,
\]

where \( \Delta d_t \) is often a function of \( d_{t-1} \) and some other inputs \( x_t \) at time \( t \).

We use the repeating cell proposed by Nascimento and Viana [14], [15] to model cumulative damage through recurrent neural networks. As illustrated in Fig. 2, “MODEL” maps the inputs \( x_t \) and previous damage \( d_{t-1} \) into a damage increment \( \Delta d_t \). 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.
B. 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 [22], the fatigue life is calculated by

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

(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 Fig. 3-(a), \( a_{SKF} \) is the SKF life modification factor, see Fig. 3-(b), \( C \) 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, \( \eta_c \) [23]). \( \kappa \) is expressed as

\[ \kappa = \frac{\nu}{\nu_1}, \]  

(4)

where \( \nu \) actual operating viscosity of the lubricant (mm²/s) [24], and \( \nu_1 \) rated viscosity (mm²/s) [25], depending on the bearing mean diameter and rotational speed.

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

\[ L_{nm}^{BRG} = \frac{t_1}{\Sigma \frac{L_{nm}^{BRG}}{L_{nm}^{BRG}}}, \]  
\[ L_{nmh}^{BRG} = \frac{t_1}{\Sigma \frac{L_{nmh}^{BRG}}{L_{nmh}^{BRG}}} \]  

(5)

where, \( t_1 \) is number of hours the turbine ran at \( N_t \) rpm. In other words, the Palmgren-Miner’s rule characterizes the incremental damage at each cycle:

\[ \Delta d_{it}^{BRG} = \frac{n_t}{L_{BRG}^{BRG}} \]  

(6)

where, the subscript \( t \) indicates the time step and \( n \) is the number of cycles per time step.

<table>
<thead>
<tr>
<th>Reliability (%)</th>
<th>Probability of failure (%)</th>
<th>( L_{nm} )</th>
<th>( a_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>10</td>
<td>( L_{10m} )</td>
<td>1.00</td>
</tr>
<tr>
<td>95</td>
<td>5</td>
<td>( L_{5m} )</td>
<td>0.62</td>
</tr>
<tr>
<td>96</td>
<td>4</td>
<td>( L_{4m} )</td>
<td>0.53</td>
</tr>
<tr>
<td>97</td>
<td>3</td>
<td>( L_{3m} )</td>
<td>0.44</td>
</tr>
<tr>
<td>98</td>
<td>2</td>
<td>( L_{2m} )</td>
<td>0.33</td>
</tr>
<tr>
<td>99</td>
<td>1</td>
<td>( L_{1m} )</td>
<td>0.21</td>
</tr>
</tbody>
</table>

(a) \( a_1 \) life adjustment factor  
(b) \( a_{SKF} \) life adjustment factor

Fig. 3 Life adjustment factors [22].

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 [24] 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 [26]. As for contamination factor [23], we considered that virgin grease would present slight contamination (as per SKF catalogue), while degraded grease would present very severe contamination (as per SKF catalogue). Fig. 4 illustrates the variation of grease properties for the virgin and degraded greases.
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 can be built such that:

\[ d_t = d_{t-1} + \Delta d_t, \]

where:

- \( d_t \) is the bearing cumulative damage, which damage increment \( \Delta d_t^{BBG} \) 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 multi-layer perceptron model.

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 \( (W_S) \) and the bearing temperature \( (T_t) \) as input variables. The cell will be recurrently used, as in Fig. 1-(a), updating both the grease and bearing damages from previous time step \( (d_{t-1}^{GRS} \) and \( d_{t-1}^{BBG} \), respectively). While \( W_S \) is mapped to equivalent dynamic bearing load \( (P_t) \) (see Fig. 6), \( T_t \) and cumulative grease damage from previous time step \( (d_{t-1}^{GRS}) \) are used to calculate grease damage parameters \( \kappa_t \) and \( \eta_{ct} \) (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 \( (\tau_f/N_f) \) for bearing fatigue damage increment \( (\Delta d_t^{BBG}) \) 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 multi-layer perceptron\(^\dagger\) using \( P_t, T_t, \) and \( d_{t-1}^{GRS} \) as inputs. The multi-layer perceptron architecture is further discussed in Section IV.D. The multi-layer perceptron 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 recurrent neural network aims at calibrating the multi-layer perceptron using grease damage observations and let it learn the damage accumulation on grease.

\(^\dagger\) The LSTM cell uses single layer perceptrons with pre-defined activation functions, shown as squares in Figure 1-(c). Here, our model uses a multi-layer perceptron.
IV. Case study

A. 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 1 provides some key parameters of wind turbine and main bearing used in our case study.

<table>
<thead>
<tr>
<th>Wind turbine</th>
<th>Rated power (MW)</th>
<th>Cut-in Wind Speed (m/s)</th>
<th>Rated Wind Speed (m/s)</th>
<th>Cut-out Wind Speed (m/s)</th>
<th>Maximum Rotor Speed (rpm)</th>
<th>Hub Height (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>3.5</td>
<td>12</td>
<td>25.0</td>
<td>20</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Main bearing</th>
<th>Designation</th>
<th>Basic dynamic load rating $C$ (kN)</th>
<th>Fatigue load limit $P_u$ (kN)</th>
<th>Mass (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKF 230/600 CAW33</td>
<td>6000</td>
<td>750</td>
<td>405</td>
<td></td>
</tr>
</tbody>
</table>

Mapping from wind speed to dynamic bearing loads is maintained using a published National Renewable Energy Laboratory (NREL) report [27], 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 Fig. 6.
B. Nominal wind speed and bearing temperature

Site-specific data is obtained from a database also provided by NREL [28], 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 Fig. 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.

Fig. 6 Wind speed mapping for case study turbine.
Fig. 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).

C. 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 Fig. 8):

\[ d_{\eta_c} = d^\eta = \frac{1}{(L_{GRS})^2} \]  

(8)

where \( d^\eta \) 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 Fig. 4):

\[ \nu = d^\eta (\nu_{deg} - \nu_{vir}) + \nu_{vir} \]  

(9)

\[ \eta_c = d_{\eta_c} (\eta_{cdeg} - \eta_{cvir}) + \eta_{cvir} \]  

(10)

where \( \nu \) and \( \eta_c \) are viscosity and contamination factor of the grease respectively.

Fig. 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 recurrent neural network model, the grease damage vector is:

\[ d_{GRS} = \{d^\eta, d_{\eta_c}\} \]  

(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 $d^c$ and $d^n c$ 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., $d^c = d^n c = 0$ after regreasing).

D. 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 III-B), 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, $\delta d^{GRS}$, is a multi-layer perceptron and the bearing damage accumulation is physics-based. Table 2 details the multi-layer perceptron architectures tested in this work. The inputs of the multi-layer 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. 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 V.

Table 2 Grease degradation increment multi-layer perceptron (MLP) models, $\delta d^{GRS}$, layer details for different architectures.

<table>
<thead>
<tr>
<th>Layer</th>
<th>MLP#1</th>
<th>MLP#2</th>
<th>MLP#3</th>
<th>MLP#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense #1</td>
<td>40 / sigmoid</td>
<td>20 / tanh</td>
<td>20 / tanh</td>
<td>10 / tanh</td>
</tr>
<tr>
<td>Dense #2</td>
<td>20 / elu</td>
<td>10 / elu</td>
<td>10 / elu</td>
<td>5 / elu</td>
</tr>
<tr>
<td>Dense #3</td>
<td>10 / elu</td>
<td>5 / tanh</td>
<td>5 / elu</td>
<td>1 / sigmoid</td>
</tr>
<tr>
<td>Dense #4</td>
<td>5 / elu</td>
<td>1 / sigmoid</td>
<td>1 / sigmoid</td>
<td></td>
</tr>
<tr>
<td>Dense #5</td>
<td>1 / sigmoid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>1,251</td>
<td>351</td>
<td>351</td>
<td>101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer</th>
<th>MLP#5</th>
<th>MLP#6</th>
<th>MLP#7</th>
<th>MLP#8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense #1</td>
<td>10 / elu</td>
<td>10 / tanh</td>
<td>5 / elu</td>
<td>2 / elu</td>
</tr>
<tr>
<td>Dense #2</td>
<td>5 / elu</td>
<td>5 / tanh</td>
<td>1 / sigmoid</td>
<td>1 / sigmoid</td>
</tr>
<tr>
<td>Dense #3</td>
<td>1 / sigmoid</td>
<td>1 / sigmoid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>101</td>
<td>101</td>
<td>26</td>
<td>11</td>
</tr>
</tbody>
</table>

The constructed multi-layer perceptron 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. 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:

$$Loss = \frac{1}{N_T N_O} \sum_{j=1}^{N_T} \sum_{t=1}^{N_O} (d_{ij}^{GRS} - \tilde{d}_{ij}^{GRS})^2$$  \hspace{1cm} (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 $\tilde{d}_{ij}^{GRS}$ 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:

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3$$  \hspace{1cm} (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 Fig. 9. In this illustration, wind speed and bearing temperature are the two inputs of multi-layer perceptron and the grease damage increment $\delta d_{GRS}$ is the output of the multi-layer 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 multi-layer 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 9: Plane approximation to actual data. In this illustration the third input variable, $d_{GRS}$, is fixed to 0.5.](image)

We initially train our multi-layer perceptron model with the plane approximation. To achieve that, we used the RMSprop§ 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) multi-layer perceptron models inside recurrent neural network 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 Chapter V Results and discussion, we show how the recurrent neural network performs when initialized with 10 different randomly generated $\alpha_i$ coefficients (all constrained by engineering judgement of how inputs affect the output).

E. Replication of results

Our implementation is all done in TensorFlow** (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 recurrent neural network using a pretrained multi-layer perceptron 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 multi-layer perceptron training, (2) trains the multi-layer perceptron with randomly generated initial weights, (3) trains the recurrent neural network using the initialized multi-layer perceptron, 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). To make use of the data and together with the codes, one only needs to download the data and extract folders inside “wind_bearing_dataset” to the directory where the “pinn_wind_bearing” repository is cloned.

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

** www.tensorflow.org
V. Results and discussion

Fig. 10 presents 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.

Fig. 11 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).

As detailed in section IV-D, we generated 10 random planes (as exemplified in Fig. 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 Fig. 12. As we expected, predictions are far away from accurate, since we randomly generated plane coefficients to approximate input-output relationship of multi-layer perceptron. However, in some cases (such as Case 1), trend of the predictions are somewhat aligned with the actual values. Fig. 12 provides a good understanding of how the initial approximations may vary from one another. While Case 1 in the Fig. 12-(a) is an example of relatively accurate initial approximation, Case 9 shown in Fig. 12-(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 recurrent neural networks for cumulative damage is that the multi-layer perceptron 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 multi-layer perceptron, even before training.

![Graph showing cumulative damage propagation and observations.](image)

**Fig. 11** All turbines $d^{GRS}$ propagation and observations.

![Graphs showing normalized outputs of randomly generated plane representations against actual output values.](image)

**Fig. 12** 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.

After the initialization of the weights, the multi-layer perceptron is integrated into the recurrent neural network framework. Fig. 13 illustrates the prediction capability of recurrent neural network 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 Fig. 13 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 initial weights of the multi-layer perceptron makes a difference here. Using same number of epochs and same optimization settings, while Case 1 almost aligns with $45^\circ$ 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 Fig. 13 implies that the performance of trained recurrent neural network 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 initialization of the multi-layer perceptron parameters.
In Fig. 14, we presented all 10 cases, where we initiated the weights randomly and trained within the recurrent neural network framework. As we can see from Fig.14, 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 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 Fig.14 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.

Finally, we implemented our trained models into our larger physics-informed neural network model 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 Fig. 16-(a)), 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.

This error in $d_{GRS}$ estimation is also reflected on the bearing fatigue damage accumulation. In Fig. 16-(b), 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.

![Fig. 15](image) Loss function variation per epoch for all cases.

![Fig. 16](image) Damage propagation for three different cases.

(a) $d_{GRS}$ propagation. (b) $d_{BRG}$ propagation.

We also studied how different multi-layer perceptron architectures (with different number of layers, neurons, and activation functions, as presented in Table 2) affect the performance of the damage accumulation model. We followed the same steps as elaborated above. Only this time, we fixed our initial plane approximation to Case 3, for the sake of providing consistency across all multi-layer perceptron models, and this way, isolating the influence of the architectural difference. In Fig. 17, we illustrated the performance of different models on the same validation data set used in the previous sections (described in section IV-B). We can infer that prediction capabilities of different architectures are not at significantly far away from one another. We can also observe the similar performance of models with different complexity levels in Fig. 18. This indicates that obtained results are dependent of the specific training configuration of the recurrent neural network (optimization parameters such as the algorithm, learning rate, number of epochs, etc.).
The grease damage accumulation is just one part of our hybrid model. Ultimately, we are interested in the impact the multi-layer perceptron models have in bearing fatigue life. For this purpose we picked the most complex multi-layer perceptron (MLP#1) and the simplest one (MLP#8) and carried out grease damage and bearing fatigue damage accumulations with these two designs. Fig. 19-(a) shows grease degradation trend has not been entirely captured by either model. Models tend to overestimate grease damage accumulation early on, and then underestimate at times as the next regreasing nears. Nevertheless, Fig. 19-(b) indicates that both models were able to predict fatigue damage propagation at acceptable accuracies. Both hybrid models slightly underestimate fatigue life. Small conservatism might not be detrimental in this application as model is likely to be updated with bearing inspection (e.g., borescope data). Difference between model predictions is within a month (out of sixteen years).

VI. Conclusions 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 multi-layer perceptron parameters 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,
- 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, and
- for this particular problem and data set, different levels of multi-layer perceptron complexity do not seem to affect the model performance significantly (as in a long term bearing fatigue prediction, deviation is about less than a month).
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 multi-layer perceptron 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.

![Diagram](image_url)

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

**Fig. 19** Damage accumulation of two designs with different complexity levels.

### 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 [29]:

$$
I_{num}^{GRS} = I_{num}^{GRS^*} K_N K_b F_1 F_2 F_3 F_4 F_5 F_6
$$

(14)

Fig. 20-(a) illustrates how grease service life varies with temperature. Most adjustment factors are given in Table 3. $F_3$ is a factor that accounts for dynamic load variation and it is shown in Fig. 20-(b). As stated by Lugt [30], 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.
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. [31]. 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:

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

where:

\[\beta_1, \beta_2, \beta_3, \beta_4\]
• $T_{BRG}$ is the bearing temperature (°K)
• $T_{Nacette}$ 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 4.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_1$</td>
<td>0.987</td>
<td></td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.0113</td>
<td></td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>0.0115</td>
<td>K s$^2$/rad$^2$</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>0.0146</td>
<td>K/MW</td>
</tr>
</tbody>
</table>

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 Fig. 6. $T_{Nacette}$ 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):

$$T_{Nacette}(t) = 0.5 T_{Ambient}(t) + 250$$

The coefficients of this equation were estimated by mapping minimum and maximum ambient temperature and main bearing temperature at the location reported by Cambron et al. [31].

D. Activation functions

In this study, sigmoid and elu activation functions are used within the multi-layer perceptron layers. These functions are given as follows and Fig. 21 illustrates these activation functions.

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

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

Fig. 21 Activation functions.

Acknowledgments

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.
Authors would like to acknowledge that they used publicly available information for this study. They understand the resulting models are simplified and that further details about design and operation of specific wind turbines would greatly improve the realism of our models (although, this information tends to be highly proprietary). Nevertheless, the models were built based on engineering judgment of available data. The intention was to build a realistic numerical study and also depart slightly from OEM-specific information. In other words, authors hope the study is realistic but have made no effort to design it in a way to reflect any specific behavior associated with any OEM data.

References


