Physics-Informed Neural Networks for Remaining Useful Life Estimation of a Vapor Recovery Unit System

Sergio Cofré-Martel[a], Enrique Lopez Droguett[b], and Mohammad Modarres[a]
[a] Department of Mechanical Engineering, Center for Risk and Reliability (CRR), University of Maryland, USA, scofre@umd.edu, modarres@umd.edu
[b] Department of Civil and Environmental Engineering, University of California Los Angeles (UCLA), USA, eald@ucla.edu

Abstract: Prognostics and health management (PHM) has become a key instrument in the reliability community. Great efforts have gone into estimating systems' remaining useful life (RUL) by taking advantage of monitoring data and data-driven models (DDMs). The latter have gained significant attention since they are model-independent and do not require previous knowledge of the system under study. This is known as black-box behavior. However, DDMs developed for PHM frameworks are commonly tested on simulated or experimental data sets, which do not present the characteristics and intricacies of data collected from monitoring sensor networks in real systems. Furthermore, the black-box behavior hinders DDMs' interpretability, and thus suffers trustworthiness, for example in maintenance decision-making processes. To address this, physics-informed models have been implemented through hybrid models, which present significant improvements in accuracy and interpretability. Particularly, physics-informed neural networks (PINNs) have been proposed in deep learning (DL) to both solve and discover a system's governing partial differential equation (PDE). This paper presents an implementation of a PINN-RUL model to a case study from a real complex engineering system (CES). The system consists of a vapor recovery unit (VRU) at an offshore oil production platform. Challenges when creating RUL labels based on maintenance logs are discussed. Results show that a trained PINN-RUL model successfully allows the interpretation of a real system's degradation dynamics through a latent variable.

1. INTRODUCTION

Prognostics and health management (PHM) has become one of the main research fields in the reliability community. As part of preventive maintenance techniques, PHM seeks to build end-to-end frameworks capable of extracting, analyzing, and processing sensor monitoring data to train diagnostics and prognostics models. These assessment models are then used to enhance maintenance policies based on an active monitoring of the assets [1]. Figure 1 illustrates an example of a PHM framework with its four principal stages: data acquisition, data preprocessing, diagnostics and prognostics model training, and maintenance decision making [2].

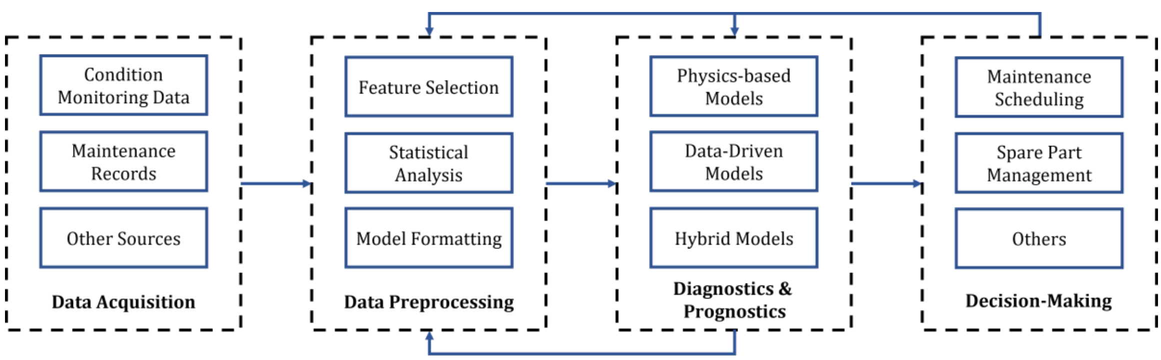

Figure 1: Example of a PHM framework.

The principal stage in any PHM framework consists of finding representative models to assess a system's state of health for diagnostics and prognostics tasks. There are mainly three types of models used to this end [3]: physics-based, data-driven, and hybrid models. Due to the great advances in data-driven models (DDMs), such as machine learning (ML) and deep learning (DL) techniques, and the lack of degradation physics-based models for complex engineering systems (CESs), most research works in PHM have focused on developing and adapting DMMs to PHM [4]. Hybrid methods, on the other hand, cover a wide range of possible techniques, where several statistical models and DDMs can be combined to obtain more accurate predictions [5], or physics models can be introduced to aid and guide the DDMs during the training process [6], [7]. In this regard, physics-informed neural networks (PINNs) have lately been introduced as an alternative to model, solve, and discover partial differential equations (PDEs) through regular neural networks (NNs) [8]. This results in networks which are cheaper to train in terms of computational costs and that are forced to follow boundary and initial conditions through a penalization function during the training process.

Applications of DDMs to PHM frameworks range from vibration-based structural damage localization and quantification [9], to degradation and fault detection in rotational elements [10]. In prognostics, most efforts have gone into estimating a system's remaining useful life (RUL). Examples of these are adaptation of convolutional neural networks (CNNs) [11], long-short term memory (LSTM) cells [12], and self-attention mechanisms [13]. However, there are several challenges that need to be addressed before implementing DL models to real scenarios and, therefore, their deployment in the industry is still rare and limited [14].

Current challenges in DL applications to PHM are thoroughly presented in [14]. Here, it is discussed that most DL techniques lack interpretability within their structure, therefore relying on post-hoc (post-model) interpretability tools rather than intrinsic (in-model) interpretability. Post-model interpretability involves methods that analyze models after the training process [15]. Examples of these are LIME (Local Interpretable Model-Agnostic Explanations) [15] and SHAP (SHapley Additive exPlanations) values [16]. Relying on external tools to gain model interpretation hinders the transparency and trust that users may have on the model's predictions [17], and is thus undesired. Another challenge identified in PHM is that most of these DL models are trained and validated using benchmark datasets generated in simulated or controlled experimental setups [2], [18]. As such, it is likely that the developed DL frameworks will present poor performance when adapted to real-world scenarios.

To address these challenges, we have previously presented a PINN framework for RUL estimation [19]. However, the framework was tested on the C-MAPSS data set [20], which although provides a good benchmark to compare models, it does not present the intricacies and challenges that can be found in sensor monitoring data collected from real CESs such as noise, missing values, and other inconsistencies [2], [21]. Further, there is a gap in the literature on how to create reliable RUL labels for prognostics tasks. Real CESs rarely present multiple failures in short periods of time and the sensor data collected are not automatically labeled. The lack of discussion on how to create RUL labels also comes as consequence of using benchmark datasets to validate PHM frameworks, since they are often provided with predetermined and well-defined RUL labels. The importance of data preprocessing in PHM and the most relevant steps to create degradation (i.e., classification) labels are presented in [2]. Nevertheless, using this methodology to generate robust and reliable RUL labels when analyzing data acquired from real CESs remains a challenge.

This paper seeks to validate the PINN-RUL framework [19] on a real case study consisting of a vapor recovery unit (VRU) located at an offshore oil production platform. The framework establishes a link between the system's recorded sensor data behavior and degradation processes through an adaptation of PINNs, estimating the RUL and providing visualization tools through a latent variable. A methology to create RUL from maitenance records is presented.

The contributions of this paper are the following:

1) This is the first implementation of PINNs for RUL estimation in a real CESs. This is a further validation of the work presented in [19].

2) We present an analysis on the RUL labels generation process based on maintenance and failure records. This is an extension of the work presented in [2].
3) We discuss how this framework can be trained in a semi-supervised manner, which is advantageous when analyzing real CESs with few failure events.

The remainder of this paper is structured as follows. Section 2 is dedicated to the case study description. Section 3 describes the PINN-RUL framework and the RUL label generation based on maintenance logs. Section 4 presents the main results for the prognostics model. Finally, conclusions are presented in Section 5.

2. VAPOR RECOVERY UNIT DATASET

The vapor recovery unit (VRU) is located at an offshore oil production platform. The VRU is a complex multi-component system responsible of recompressing hydrocarbons separated from the oil stream during the primary processing stages. The main components integrating the VRU system are two scrubbers, two heat exchangers, an electric motor, and a two-stage screw compressor. Figure 2 shows a sketch of the layout of the system.

The VRU unit is equipped with a sensor network comprised of 189 sensors of different kinds such as flowmeters, thermocouples, pressure gauges, and accelerometers. Sensor data logs are registered at a sampling interval of 15 sec. from 1 January 2019, until 18 February 2020. The sensor network is composed of multiple redundant sensors and thus, many variables are highly correlated. The collected data comes from a real-world operational environment; therefore, it presents noise contamination and missing data logs, among other defects. Along with the sensor readings, a separate file is provided with information regarding the stoppages of the system. Here, for each time the system was stopped, a label and an observation describing the cause of the shutdown are registered. Thus, the failure times for each of the components are available through the maintenance logs, as well as every time the system was paused or shut down for external reasons.

A detailed description of the system and the preprocessing of the data can be found in [2], which considers temporal and statistical analysis for feature selection, outlier detection and removal, as well as training and test set separation.

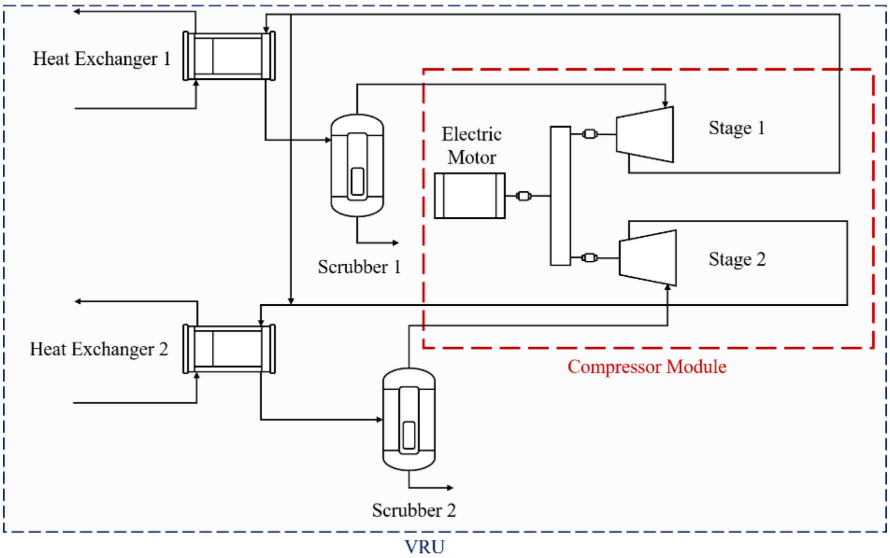

Figure 2: Simplified Diagram of Vapor Recovery Unit.

3. RUL LABELS AND PINN FRAMEWORK

This section provides details on the RUL label generation process based on maintenance logs and the PINN-RUL framework.

3.1 RUL Label Generation

The proposed RUL label generation process for components in a multi-component system follows a similar philosophy as the one presented in [2]. Maintenance logs are used to identify stoppage times related to failures. In this case, the mechanical failures associated to the scrubber component are considered. The failure times determine where a time window is considered to create the RUL labels.

Formally, the RUL labels are created as follows. Let S be a multi-component system composed of N components. Each component is then denoted as C_i, with $i = \{1, \dots, N\}$, which has a set of failure modes FM_j^i with $j = \{1, \dots, n_i\}$, where n_i is the number of possible failure modes for the i-*th* component. Finally, let t_i^j denote the time at which the system was stopped due to failure mode j at the i-*th* component, and t_S^p as the time at which the system is stopped or paused for any reason that is not one of the known failure modes.

When a component C_i fails at failure time t_i^j, it can be assumed that this corresponds to $RUL_i = 0$. Starting from here, a time window $\Delta\, t_w$ can be define in which the RUL of the component will decrease linearly in time [11]. That is, at time $t_j^j - \Delta t_w$ the component has a $RUL_i = \Delta t_w$ label. Figure 3 illustrates an example of this methodology for a time-window $\Delta\, t_w = 1400$min. Setting a defined time window before the failure event is critical to obtain robust RUL labels. Indeed, if all the monitoring sensor data between one failure and the next were used, it is likely that the data will be contaminated with other non-failure events (e.g., maintenance, pauses, emergency shutdowns). By defining Δt_w, the generated labels are then directly linked to the failure mode of the component and are defined based on the failure FM_j^i of the component and not the entire system.

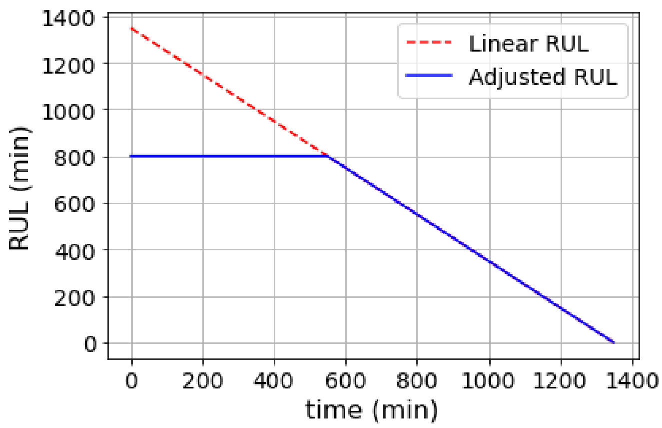

Figure 3: Example of RUL label generation.

As it is discussed in the literature [11], when defining RUL labels based on time-to-failure one could choose to use a linear RUL label until the failure event, or select a start of degradation point from where the RUL decreases linearly in time. Figure 3 shows an example for both approaches, where the red line illustrates the linear RUL decrement while the blue line is the adjusted RUL when the start of degradation point is selected at 800 min. before the failure. That is, it is assumed that the RUL is constant until 800 min. before the failure, which indicates the start of the degradation process.

For this case study, the selected time window is set for 24 hrs. (i.e., $\Delta t_w = 1440$ min.) to create the RUL labels for the scrubber component. It is important to ensure that no other pauses or stoppages overlap with the selected time window. That is, $t_s^p \notin \left(t_i^j - \Delta t_w; t_i^j\right)$.

3.2 PINN-RUL framework

The PINN-based framework is illustrated in Figure 4 [19]. The framework consists of three stages, each represented by a NN. The first stage maps the operational conditions (OC, i.e., sensor values) and the time t into a two-dimensional latent variable x. The time variable represents how far into the future the RUL estimation is made. For instance, for $t = 0$ the prediction would correspond to the RUL at the time the sensor values were obtained. On the other hand, if $t = 10$ min. the estimated RUL would correspond to a prediction 10 min. in the future starting from the time at which the OC were observed. That is, the framework allows to use current operational conditions to predict RUL values at future instances in time.

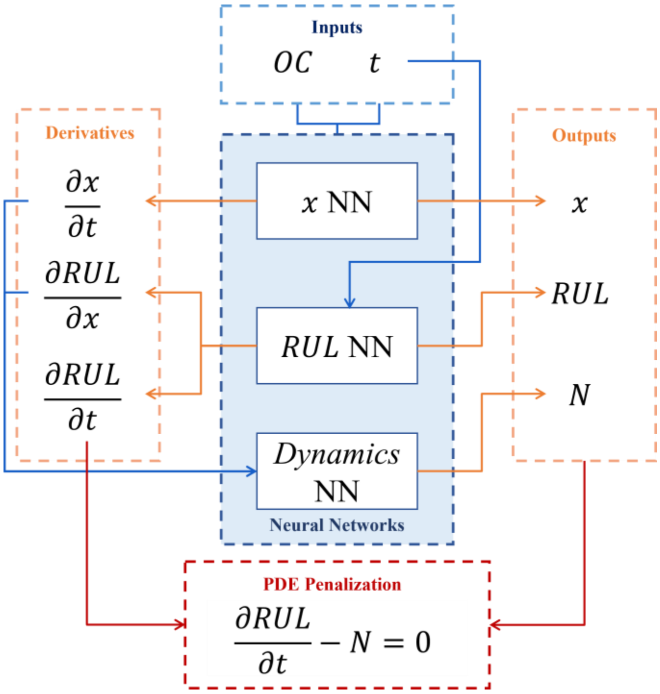

Figure 4: Physics-informed neural network RUL framework.

The framework's second stage is the most similar to other DL frameworks implemented for RUL estimation. Here, the output x from the first stage and time t are used as input values to the second NN that outputs the system's RUL. The time is considered in this stage so that automatic differentiation [22] can be used to obtain the exact derivative of RUL w.r.t time ($\partial RUL/\partial t$), which is then used as the left-hand side of the PDE penalization function. The third stage of the framework corresponds to the dynamic NN as defined in [8] and is used as the right-hand side of the PDE penalization. This NN takes x and the derivative $\partial RUL/\partial x$ as input values, and thus the PDE penalization f is defined as:

$$f := \frac{\partial RUL}{\partial t} - N\left(x, \frac{\partial RUL}{\partial x}\right) = 0 \tag{1}$$

where N corresponds to the output of the dynamic NN. Notice that the dynamic NN does not need any labels to be trained and could potentially be trained in an unsupervised manner. As usual, the training loss for the second stage (i.e., the RUL) corresponds to the mean squared error defined as:

$$MSE = \frac{1}{M} \sum\nolimits_{i=1}^{M} (RUL_i - y_i)^2 \tag{2}$$

where M is the number of training samples, y_i is the RUL label, and RUL_i is the corresponding prediction from the second stage. The cost function can then be defined as:

$$Cost = MSE + \lambda f^2 \tag{3}$$

where λ is the penalization weight.

4. RESULTS AND DISCUSSION

In this section, results are presented for the PINN-RUL models trained on the VRU system's scrubber component. A 10-fold cross validation is implemented for the selection of the main hyperparameters, namely: number of layers and neurons in each NN, activation functions, PDE penalization weigh, and learning rate. More details on the hyperparameters selection can be found in [19]. Models are trained using Python 3.7 using the Tensorflow 2.0 library [23]. An Intel i7-9700K CPU and a 24GB Titan RTX GPU are used as hardware. The average training of a model takes an average of 36.1 min. The long training times are mostly due to the size of the training data set, corresponding to 2M data points.

Once the hyperparameters are selected, 10 different models are trained, which yield an average RMSE of 2.18 hrs. for the training set, and 2.19 hrs. for the test set. Figure 5 shows the latent variable 2D representation for one of the trained models, which is color mapped with the corresponding label (left) and estimated (right) RUL value. A transition from concentrated high RUL values (healthy) to disperse low (degraded) RUL values is observed. On the one hand, the dispersion of low RUL values could correspond to different failure mechanisms. In the case of the VRU's scrubber, information on the failure mechanism is not available, but it is likely that more than one failure mode is at play.

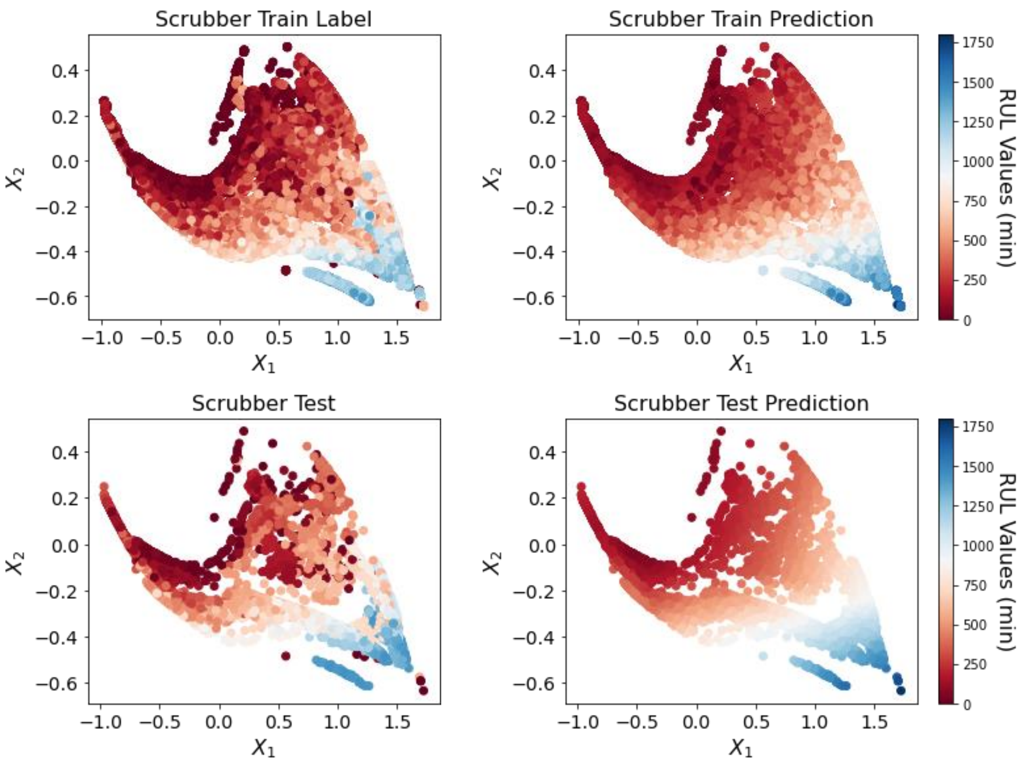

Figure 5: Latent variable RUL mapping for the VRU Scrubber.

On the other hand, having high RUL values concentrated in a small area can be interpreted as the system behaving consistently when is in a healthy state. Figure 5 also shows that the model smooths the transition from high RUL values to low RUL values labels. When evaluating the test set, Figure 5 shows that the latent variable mapping follows a similar pattern as the train set. As such, the latent variable map gives an accurate assessment of relative position of a new incoming data with respect to the training set, which in turn provides interpretability on the system's state of health.

The obtained results are important since they show that it is possible to obtain RUL estimation based on big machinery data from a real CES. The implementation of PINNs also provides the latent space representation, which can make an important difference in the decision-making process by offering more transparency and trust on the obtained RUL estimation. Indeed, an online implementation of this model would yield a simultaneous RUL value and its location on the latent space. If the RUL values does not correlate with the latent variable's color map, it is likely to be an inaccurate prediction. Further, as it is presented in [19], health state classifiers can be trained by selecting a RUL threshold.

One of the advantages of the PINN-RUL framework is that it does not require all training data to be labeled. Indeed, although RUL labels are needed to train the RUL NN (second stage in Figure 4) and, therefore, the entire framework, unlabeled data points can still be used to feed the penalization function f in Equation 3. This implies that a model can be trained in a semi-supervised way in the case that labels are not available (e.g., the system does not present failures), or the RUL labels are not reliable. The latter is important when considering the proposed methodology to create the RUL labels in Section 3.1. Considering a time window only before the failure event to create the RUL labels can result in most of the data set not been used to train the model, particularly when Δt_w is small or there are few failure events registered in the available data. Thus, feeding unlabeled data to the penalization function during the training process can add valuable information to the trained model, especially for the healthy state of the system (i.e., high RUL values).

It should be noted that the RUL label generation methodology is limited by the number of failures registered in the maintenance logs, as well as the quality of the registered data. Ideally maintenance logs should contain exact time of failure and accurate descriptions on the failure mechanisms that caused each failure. Furthermore, the selected time window Δt_w will depend on the system and component under study. It is well known that DDMs models tend to perform worse for high RUL values than for low RUL values. This can be due to overrepresentation of the low RUL labels, since all equipment fail, but not all have long operational times before the next failure. This is a factor that needs to be considered when choosing the Δt_w value, since high values would likely result in an underperformance of the model.

The model's interpretability provided by the latent variable x in Figure 5 can be considered as intrinsic since it comes as a result from the model constraint (i.e., PINN penalization function) imposed during the model training [17]. Interpretability is an ongoing challenge in the ML and artificial intelligence (AI) community and not just in PHM where it has been identified as one of the most important challenges to deploy DDM-PHM frameworks in real systems [14]. Hence, it is important that new proposed DL-PHM frameworks give means to bring in more transparency and understanding of their predictions, and the presented PINN-RUL takes a step forward in that direction.

5. CONCLUSIONS

This paper presents and validates an adapted PINN framework for RUL estimation of a VRU system. The generation of RUL labels based on maintenance logs is also presented. The trained models provide an intrinsic interpretation of the system's state of health through a latent variable. Future work should consider aggregating the model's results to the decision-making process (i.e., maintenance policy design) by creating metrics and elements to incorporate the RUL estimation and the diagnostic latent variable. Additionally, new Bayesian physics-informed neural networks (B-PINN) [24] could become an important tool to quantify the model's uncertainty.

ACKNOWLEDGEMENTS

Sergio Cofre-Martel would like to thank the Agencia Nacional de Investigación y Desarrollo (ANID-Doctorados Becas Chile-72190097) for their support through the Becas Chile program.

REFERENCES

[1] N. Omri, Z. Al Masry, N. Mairot, S. Giampiccolo, and N. Zerhouni, "Towards an adapted PHM approach: Data quality requirements methodology for fault detection applications," *Comput. Ind.*, vol. 127, p. 103414, May 2021, doi: 10.1016/j.compind.2021.103414.

[2] S. Cofre-Martel *et al.*, "Big Machinery Data Preprocessing Methodology for Data-Driven Models in Prognostics and Health Management," *Sensors*, vol. 21, no. 20, p. 6841, Oct. 2021, doi: 10.3390/s21206841.

[3] B. Rezaeianjouybari and Y. Shang, "Deep learning for prognostics and health management: State of the art, challenges, and opportunities," *Meas. J. Int. Meas. Confed.*, vol. 163, p. 107929, 2020, doi: 10.1016/j.measurement.2020.107929.

[4] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, Jan. 2019, doi: 10.1016/j.ymssp.2018.05.050.

[5] M. Soleimani, F. Campean, and D. Neagu, "Integration of Hidden Markov Modelling and Bayesian Network for fault detection and prediction of complex engineered systems," *Reliab. Eng. Syst. Saf.*, vol. 215, no. May, p. 107808, 2021, doi: 10.1016/j.ress.2021.107808.

[6] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021, doi: 10.1038/s42254-021-00314-5.

[7] S. F. Karimian, R. Moradi, S. Cofre-Martel, K. M. Groth, and M. Modarres, "Neural Network and Particle Filtering: A Hybrid Framework for Crack Propagation Prediction," pp. 1–12, 2020, [Online]. Available: http://arxiv.org/abs/2004.13556.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.

[9] S. Cofre-Martel, P. Kobrich, E. Lopez Droguett, and V. Meruane, "Deep Convolutional Neural Network-Based Structural Damage Localization and Quantification Using Transmissibility Data," *Shock Vib.*, vol. 2019, pp. 1–27, Sep. 2019, doi: 10.1155/2019/9859281.

[10] M. Cerrada *et al.*, "A review on data-driven fault severity assessment in rolling bearings," *Mech. Syst. Signal Process.*, vol. 99, pp. 169–196, 2018, doi: 10.1016/j.ymssp.2017.06.012.

[11] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliab. Eng. Syst. Saf.*, vol. 172, no. October 2017, pp. 1–11, 2018, doi: 10.1016/j.ress.2017.11.021.

[12] A. Elsheikh, S. Yacout, and M. S. Ouali, "Bidirectional handshaking LSTM for remaining useful life prediction," *Neurocomputing*, vol. 323, no. October 2018, pp. 148–156, 2019, doi: 10.1016/j.neucom.2018.09.076.

[13] J. Xia, Y. Feng, C. Lu, C. Fei, and X. Xue, "LSTM-based multi-layer self-attention method for remaining useful life estimation of mechanical systems," *Eng. Fail. Anal.*, vol. 125, p. 105385, Jul. 2021, doi: 10.1016/j.engfailanal.2021.105385.

[14] O. Fink, Q. Wang, M. Svensén, P. Dersin, W. J. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Eng. Appl. Artif. Intell.*, vol. 92, no. January, p. 103678, 2020, doi: 10.1016/j.engappai.2020.103678.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 1135–1144, 2016, doi: 10.1145/2939672.2939778.

[16] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Adv.*

Neural Inf. Process. Syst., vol. 2017-Decem, no. Section 2, pp. 4766–4775, 2017.
[17] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, “Machine Learning Interpretability: A Survey on Methods and Metrics,” *Electronics*, vol. 8, no. 8, p. 832, Jul. 2019, doi: 10.3390/electronics8080832.
[18] Y. Wen, M. Fashiar Rahman, H. Xu, and T.-L. B. Tseng, “Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective,” *Measurement*, vol. 187, no. July 2021, p. 110276, Jan. 2022, doi: 10.1016/j.measurement.2021.110276.
[19] S. Cofre-Martel, E. Lopez Droguett, and M. Modarres, “Remaining Useful Life Estimation through Deep Learning Partial Differential Equation Models: A Framework for Degradation Dynamics Interpretation Using Latent Variables,” *Shock Vib.*, vol. 2021, pp. 1–15, May 2021, doi: 10.1155/2021/9937846.
[20] A. Saxena, K. Goebel, D. Simon, and N. Eklund, “Damage propagation modeling for aircraft engine run-to-failure simulation,” in *2008 International Conference on Prognostics and Health Management*, Oct. 2008, pp. 1–9, doi: 10.1109/PHM.2008.4711414.
[21] N. Omri, Z. Al Masry, N. Mairot, S. Giampiccolo, and N. Zerhouni, “Industrial data management strategy towards an SME-oriented PHM,” *J. Manuf. Syst.*, vol. 56, no. April, pp. 23–36, Jul. 2020, doi: 10.1016/j.jmsy.2020.04.002.
[22] A. Otto, K. Griewank, and A. Griewank, “On Automatic Differentiation,” 1988.
[23] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” 2016.
[24] L. Yang, X. Meng, and G. E. Karniadakis, “B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data,” *J. Comput. Phys.*, vol. 425, p. 109913, Jan. 2021, doi: 10.1016/j.jcp.2020.109913.