Physics Informed Machine Learning (PIML) methods for estimating the remaining useful lifetime (RUL) of aircraft engines

Sriram Nagaraj Southwest Research Institute (SwRI) San Antonio, Texas, USA sriram.nagaraj@swri.org Truman Hickok

Southwest Research Institute (SwRI)

San Antonio, Texas, USA

truman.hickok@swri.org

Abstract—This paper is aimed at using the newly developing field of physics informed machine learning (PIML) to develop models for predicting the remaining useful lifetime (RUL) aircraft engines. We consider the well-known benchmark NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) data as the main data for this paper, which consists of sensor outputs in a variety of different operating modes. C-MAPSS is a well-studied dataset with much existing work in the literature that address RUL prediction with classical and deep learning methods. In the absence of published empirical physical laws governing the C-MAPSS data, our approach first uses stochastic methods to estimate the governing physics models from the noisy time series data. In our approach, we model the various sensor readings as being governed by stochastic differential equations, and we estimate the corresponding transition density mean and variance functions of the underlying processes. We then augment LSTM (long-short term memory) models with the learned mean and variance functions during training and inferencing. Our PIML based approach is different from previous methods, and we use the data to first learn the physics. Our results indicate that PIML discovery and solutions methods are well suited for this problem and outperform previous data-only deep learning methods for this data set and task. Moreover, the framework developed herein is flexible, and can be adapted to other situations (other sensor modalities or combined multi-physics environments), including cases where the underlying physics is only partially observed or

Index Terms—Predictive maintenance, Physics Informed Machine Learning (PIML), Deep Learning

I. INTRODUCTION

Aircraft maintenance is a a critical component in ensuring the safety, dependability, and efficiency of aviation systems and their operational modes ([1]–[4]). Classical maintenance methods, such as scheduled maintenance, aimed at corrective repair/upkeep may still be relevant but are often plagued by limitations such as operational (including logistic) disruptions, periods of unavailability (downtime) and increased costs, for even routine issues. On the other hand, predictive maintenance (PM), which refers to maintenance approaches relying on data-driven or data-augmented techniques, is a proactive set of approaches that leverage past data (and modes) to foresee system (or subsystem) failures prior to occurrence ([5], [6]).

Work performed in Southwest Research Institute (SwRI)

Quantities of interest that may be part of a PM diagnosis are probability of engine failure/stalls, estimating remaining useful lifetime (RUL) of engines, probability of catastrophic multi-system failure etc. A characteristic of PM systems are real-time (or near real-time) information about the system and dynamic updating of data to provide estimates when the mission or flight is in progress. Moreover, these systems can be incorporated into the flight system itself and not only in separate command centers.

A. Salient Features:

Other important points to note about PM are:

- Since PM systems are (near) real-time, they allow for early detection and identification of aircraft degradation or faulty functioning of aircraft subsystems. In the case of commercial aircraft, this allows to potentially save passenger lives by reducing the risk of dangerous malfunction being unnoticed. PM allows for corrective, riskmitigation maneuvers on the part of pilots ([5]).
- When early onset of faulty system performance is picked by PM diagnosis, this allows for action to be taken that would help avoid further catastrophic outcomes such as complete engine failure. This would help not only saving human life, but also reduce total maintenance costs of the mission/flight ([3], [4]).
- PM allows for addressing maintenance needs during both scheduled and unscheduled periods. This in turn minimizes any unplanned downtime of aircraft systems and helps to optimize aircraft utilization.
- Continuing the earlier remarks, early detection of system malfunction allows for the use of maintenance resources for mission-critical tasks. Thus, PM ensures the focusing of in-depth maintenance efforts on aircraft/avionics systems or components that are known to require attention, thereby optimizing the allotment of maintenance resources. Finally, this helps to reduce costs overall in maintenance.
- As another means of reducing overall costs, PM reduces the chances of unexpected maintenance issues, and thereby reduces operational delays (such as flight

delays, cancellation or diversions) which are known to impact commercial airlines significantly. For military aircraft, operational delays may impact the efficacy of the mission, and may even lead to a critical mission being compromised ([5]).

- PM and ML/AI models: PM relies on innovative use of sensor data from various aircraft systems, such as engines, avionics, and airframes. These sensor readings are fused via ML/AI techniques and processed to predict anomalous behavior, potential known or unknown failures; further, in non-critical scenarios, the ML/AI methods are used to monitor equipment health. Simply put, the ML/AI techniques are trained on a large swathe of past sensor data in order to detect/identify trends and risk factors that are indicative of equipment failures or system degradation. Once detected or identified, information about these issues can be communicated to a subject matter expert (SME) to determine the severity of the issues and what corrective actions need to be taken ([2]).
- Regulatory agencies such as the Federal Aviation Administration (FAA) and the European Union Aviation Safety Agency (EASA), have developed standards and regulations that need to be met by commercial aircraft in order for them to be flight worthy. PM techniques can be used to augment existing risk management methods to further ensure compliance with these regulatory standards.

There are several challenges to full-scale incorporation of PM methods. First, data quality/integrity and interoperability challenges often prevent the implementation of PM methods. Scarce data or uneven data between different sensors limit the applicability of PM techniques. Integrating real-time systems on platform is a separate challenge, as the existing hardware and infrastructure on aircraft needs to be sufficient for running PM diagnostics. Finally, reliability of PM methods requires extensive testing, and both commercial and military operators need to allocate sufficient resources to adequate testing. However, these challenges can be surmounted with careful analysis and a well-defined systems engineering approach during design and development. The future directions of a PM based fleet are to include internet of things (IoT) devices into PM data sources and the design of digital twins to further improve the efficacy of PM.

PM is therefore an emerging, and continually evolving paradigm in aircraft maintenance practices ([1]). PM techniques can be incorporated with other evolutions in the aviation industry with newer hardware/sensing capabilities. PM can then co-evolve to ensure the safety and reliability of ever increasing complex aircraft systems. Moreover, the financial savings afforded by PM can be used to further increase aircraft capabilities.

B. Physics Informed Machine Learning

Currently, a large amount of computational modeling is based on deterministic models and/or multi-fidelity approximate solutions that do not directly account for inherent variability (stochasticity) in the material properties. When predicting real-world expected performances, these models are laboriously recomputed many times and deliver solutions that do not generalize efficiently to situations where the data needed for simulations is partially observed. Moreover, real experimental test data is a scarce commodity, so there is a need to utilize such data in the most efficient way possible ([7]).

1) PIML in a Nutshell: Physics Informed Machine Learning (PIML) models offer a promising way forward for balancing computational fidelity and data-driven generalization. PIML is a data driven ML modeling framework that is augmented by the underlying scientific domain (mechanics, electromagnetics, acoustics, thermal or multiphysics). Thus, scientific knowledge captured by some relevant set of physical principle/equation (conservation laws, Maxwell equations, mechanical properties, heat diffusion etc.) is directly integrated with the ML modeling to predict quantities of interest (failure of materials, energy transmission efficiency, stress, drag etc.) which are often needed for mission-critical decision making ([7], [8]). PIML models achieve better generalization than traditional "data only" ML models since they can be viewed as incorporating real-world prior knowledge (i.e. a Bayesian reformulation). This also allows PIML models to operate in low/partially observed data regimes where the physicsinformed component can act as data surrogate as needed. Due to inherent regularization, PIML models are less prone to spurious solutions while dealing with noisy data as well. Finally, PIML models yield more succinct models characterized by fewer ML parameters while achieving (often exceeding) the accuracy/efficiency of complex "data only" deep learning models. This also renders then more explainable than "black box" ML models and is significant when downstream decision making requires explainable predictions ([9], [10]).

- 2) Methodologies in Physics-Informed Machine Learning: There are several ways in which PIML performed.
 - Differential Equation Constraints: PIML frameworks utilize differential equations to incorporate dynamic system behavior into machine learning models, facilitating timeseries forecasting, and dynamical system identification ([10]).
 - Generative Data: Systems governed by stochastic differential equation (SDE) models can be used to perform physics-informed data generation. This generated data would capture well the underlying physics, and being a data augmentation method, can directly be incorporated into the ML modeling to ensure model robustness ([11]).
 - Hybrid Physics-Data Models: Hybrid approaches are also possible which integrate physics-based simulations with data-driven (ML) techniques, balancing model accuracy and computational efficiency ([9]).
- 3) Modes of PIML Operation: Roughly, PIML methods operate in two modes. The first is the "discovery mode" wherein the physical parameters (coefficients of differential equations, structural parameters etc.) are learning from the underlying data. The second is the "solution mode" which

uses a fully specified physical model along with a data-driven component to drive predictions. The exact means by which the two modes operate are largely problem/domain dependent, and specific implementations of either mode of operation may change widely across different areas ([8], [11]).

4) Physics Informed Neural Networks (PINNs): PINNs are an exemplary class of PIML models. PINNs are deep neural networks which are trained on a loss functions composed of two components ([8]). The first is the usual data driven (often mean square error MSE), while the other component is a physics-informed one. The physics informed loss includes all physical constraints of the problem include modeling choices, differential/integral equations, boundary conditions, uncertain coefficients etc. Both components of the overall loss work in tandem to deliver a robust model. While PINNs are one example of a class of PIML models, it is possible to have other PIML approaches as well. In this regard, PIML is a modeling framework, rather than a specific set of model types.

5) Applications of PIML:

- Aerospace Engineering: PIML methods are very useful for turbulence modeling, wing design and shape optimization, where traditional computational methods such as finite element/finite difference methods can be augmented by data-driven models ([9], [12]).
- Mechanical Engineering: Applications in mechanical engineering includes modeling of fatigue of materials, CFD, elasticity etc. ([9], [13], [14])
- Material Science: PIML accelerates materials discovery by integrating physics-based simulations with high-throughput experimentation, guiding the design of novel materials with desired properties ([9], [15]).
- Biomedical Engineering: PIML aids in medical image analysis, patient monitoring, and drug discovery, leveraging physics-based models and clinical data to improve diagnosis and treatment outcomes ([16]-[18]).

Scalability of PIML methods to high-dimensional data remains a challenge. Lower order modeling, model order reduction methods, and surrogate models have proven to be promising ways of mitigating this challenge ([19]). Another area requiring further investigation is uncertainty quantification (UQ) of PIML predictions. UQ is needed for meaningful decision-making and risk assessment, and robust methods for uncertainty quantification and propagation are essential. Finally, software packages for easy implementation of differential equation solvers using deep learning methods are available as well ([20]).

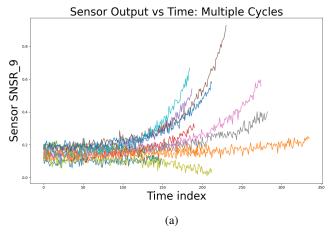
C. Prior Work: Data-Driven Deep Learning for PM using the C-MAPSS Dataset

In recent years, machine learning techniques, particularly long short-term memory (LSTM) and convolutional neural networks (CNNs), have shown promise in PM applications. Towards this end, the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset (a widely used benchmark dataset for prognostics and health management research) is a standard dataset used for studying the efficiency of ML

models for PM ([21], [22]). C-MAPSS consists of multiple simulated engine run-to-failure trajectories, each containing sensor measurements and corresponding remaining useful life (RUL) labels. The data consists of multiple noisy time series that correspond to sensor readings from different parts of an aircraft engine. Thus far, the LSTM Networks and CNNs have been well-studied architectures for PM usinng the C-MAPSS dataset ([23]–[26]). LSTMs are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. LSTMs consist of memory cells, input and forget gates, and output gates, allowing them to retain information over long sequences and adapt to varying temporal patterns. CNNs on the other hand, are deep learning architectures designed for feature extraction from structured data, such as images or time-series signals. CNNs consist of convolutional layers, pooling layers, and fully connected layers, enabling hierarchical feature learning and pattern recognition. CNNs are applied to sensor data for fault detection, anomaly detection, and condition monitoring, leveraging their ability to automatically learn relevant features from raw data. LSTMs have been successfully applied to time-series data for equipment health monitoring, remaining useful life estimation, and fault diagnosis. CNNs have been applied to sensor data for fault detection, anomaly detection, and condition monitoring, leveraging their ability to automatically learn relevant features from raw data. However, these methods are purely data-driven and do not incorporate the physics underlying the data.

D. Our Contributions:

We consider the C-MAPSS dataset as the main data for this paper. As we have seen, this is a well-studied dataset, and much work exists in the literature that address RUL prediction on C-MAPSS data with classical and data-driven (deep) ML methods. Our PIML based approach is different from previous methods, and we use the data to first learn the physics. In the absence of published empirical physical laws governing the C-MAPSS data, our approach first uses stochastic PIML methods to estimate the governing stochastic physics models from the noisy time series data (this is referred to as the "discovery" mode of PIML). In this phase, we model the sensor data as being governed by a system of first order stochastic differential equation (SDE) and estimate the mean and variance functions. When the underlying statistical distribution at a given timestep is known to be unimodal, we can use the sample mean and variance as estimates of the mean and variance functions at that timestep, or equivalently, the estimate of the K=1-mean cluster median and variance. If the distribution is multimodal (with K-modes) we use K-means clustering estimation to define the estimated mean and variance functions at the timestep as the K-medians of the collection of sample paths at the timestep in question. We are thereby able to obtain a surrogate for the underlying physics. K-means estimation is a well-established nonparametric method, and using it, we are able to model multi-modal sensor output wherein the statistical distribution of the same sensor output is a multimodal distribution (see Figure 1b and Figure 2b). We then



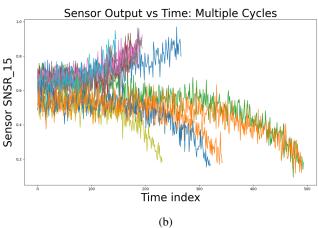


Fig. 1: (a,b) Sample sensor outputs for two different modes of operations (FD001 and FD003 respectively). (a) is well modeled as geometric brownian motion, while (b) shows multi-modal behavior of sensor output.

use existing ML models (LSTM) with the learned physics to generate predictions of RUL (the "solution" mode of PIML). Unlike traditional PIML methods, in the solution mode, we directly incorporate the mean and variance functions as data augmentation rather than indirectly through a loss function. The proposed set of methods delivers reliable, robust solutions to the challenging problems of physics discovery and physics informed prediction, relying on the superior generalization capabilities of modern deep machine learning. Our results indicate that PIML discovery and solutions methods are well suited for this problem and outperform previous data-only deep learning methods for this data set and task. Moreover, the framework developed herein is flexible, and can be adapted to other situations (other sensor modalities or combined multiphysics environments), including cases where the underlying physics is only partially observed or known.

II. STOCHASTIC PHYSICS MODEL

We now describe our physics model for the different datasets. Recall the absence of existing physics based models for the data. We therefore appeal to the approach of *learning* the physics from the raw data. To this end, we model the output S(t) of each (viable, information bearing) sensor via a stochastic differential equation (SDE). Note that, inspecting the sample paths of the sensors from FD001-FD004 (see section III for more details on the dataset) shows that this is a reasonable modeling choice. Indeed, several sensors are well-modeled as geometric brownian motion. We assume the following form for each sensor output S:

$$dS(t) = a(t)dt + b(t)dW(t), (1)$$

where W(t) is a one-dimensional martingale stochastic process (not necessarily Gaussian, but with independent increments) and a(t), b(t) are the drift and diffusion coefficients respectively. The stochastic integrals are assumed to be in the sense of Ito. Since S(t) is a stochastic process, we will, as needed, explicitly denote the dependence on a random outcome in the event space ω as $S(t,\omega)$. In reality, the drift and diffusion functions are themselves dependent on both tas well as S(t), i.e., $a(t) = \tilde{a}(t, S(t))$ and $b(t) = \tilde{b}(t, S(t))$, although for our purposes, we shall denote them as functions of t alone, with the dependence on S(t) being implicit. Thus, each data set consists of multiple cycles of several sensor values, $S_i(t)$ with i = 1, ..., N. One possible approach would be to estimate the corresponding $a_i(t), b_i(t)$, from the data directly to fully specified the SDE model. However, we take a slightly different approach. We instead estimate the mean and variance process defined by:

$$\mu(t) = \mathbb{E}[S(t)],\tag{2}$$

$$\rho(t) = \mathbb{E}[|S(t) - \mu(t)|^2]. \tag{3}$$

These are the instantaneous mean and variance functions of S(t). Note that if p(t,x) denotes the density of the process $S(t,\omega)$, i.e., $S(t,\cdot) \sim p(t,x)$ we have that $\mu(t) =$ $\int xp(t,x)dx$ and $\rho(t) = \int (x-\mu(t))^2p(t,x)dx$. We take this approach for the following reasons. First, an inspection of sensor data as a function of time (over multiple cycles) indicates that several sensors have non-Gaussian distribution at different time instances (see for e.g. Figure 1b). Typical use cases of the standard diffusive Ito SDE assume Gaussian, independent increment processes, which would not be valid in the current case. Second, estimating the coefficients a(t), b(t)directly from data would need further assumptions on the underlying process S(t), which, again, may not be valid. Instead, we stay with the most general assumptions possible, and estimate the mean and variance functions instead. Here, the only assumption being made is that the process has finite first and second moments, which is evidently true.

A. Estimating $\mu(t), \rho(t)$

We now discuss the estimation of $\mu(t)$, $\rho(t)$. We first assume that the distribution of $S(t,\omega)$ is unimodal for all t. This is

true for several sensors in the dataset. For a given sample path $S_i(t,\omega_k)$, we define the support of the path as

$$\rho_{S_i}(k) := \{t > 0 : |S_i(t, \omega_k)| > 0\}. \tag{4}$$

We assume that $\rho_{S_i}(k) = [0, T_{S_i}(k)]$ for some $T_{S_i}(k) > 0$. We set $T_{max} = \sup_{\omega_k, S_i} (T_{S_i}(k))$ to be the maximum length of the vector of any sensor's readings over all cycles. If a given sample path $S_i(t, \omega_k)$ ends before T_{max} , we extend the sample path by zero, i.e., we set $S_i(t, \omega_k) = 0, T_{S_i}(k) < t < T_{max}$.

Now, given a fixed partition $\Delta = \{0 < t_1, \dots, t_n = T_{max}\},\$ for each t_k we define a random variable as $\hat{S}_i(k,\omega)$ $S_i(t_k,\omega)$. That is, $\hat{S}_i(k,\cdot)$ is obtained by collecting the sample paths $S_i(t,\cdot)$ and evaluating them at $t=t_k$, thereby resulting in the random variable $\tilde{S}_i(k,\cdot):\Omega\to\mathbb{R}$. We then define $\hat{\mu}_i(k) = \mathbb{E}(\tilde{S}_i(k,\cdot)), \text{ and } \hat{\rho}_i(k) = \mathbb{E}(\tilde{S}_i(k,\cdot)^2) - \hat{\mu}_i(k)^2 \text{ to}$ be the mean and variance of the random variable $S_i(k,\cdot)$. Finally, we can define the estimates of the drift and diffusion functions via an interpolation operator $\Pi: \mathbb{R}^n \to C([0, T_{max}])$ as $\hat{\mu}_i(t) = \Pi(\{\hat{\mu}_i(k), k = 1, \dots, n\})$ and likewise $\hat{\rho}_i(t) =$ $\Pi(\{\hat{\rho}_i(k), k = 1, \dots, n\}).$

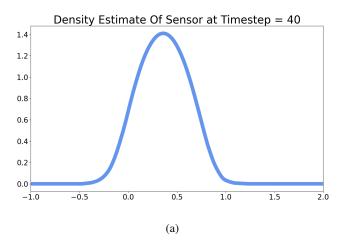
In practise, we shall be in need of only the values $\hat{\mu}_i(k)$, k= $1, \ldots, n$ (resp. $\hat{\rho}_i(k)$), and hence will not specify the exact form of the interpolation operator $\Pi(\cdot)$. Indeed, we preprocess the sensor data and compute the drift and diffusion functions using an appropriately defined histogram (or density estimate) of the random variable $S_i(k,\omega)$ at each time instant $t_k \in \Delta$. Computing $\hat{\mu}_i(k)$ is done using K- means clustering. Indeed, since we know the distribution is unimodal, we compute the single centroid of the distribution of $\tilde{S}_i(k,\cdot)$ and define it to be the mean $\hat{\mu}_i(k)$. This way, we avoid explicit computation of the expectation.

1) Estimating $\mu(t)$, $\rho(t)$: Multimodal Case: Thus far, we have tacitly assumed that the histogram (density estimate) of $\hat{S}_i(k,\omega)$ is unimodal when computing the drift and diffusion functions. In some cases, this is not true, and we need to modify our approach to include multi-modal densities. In such cases, we resort to modeling the multi-modal density of $\hat{S}_i(k,\omega)$ using a K-means estimate of the mean. Typically, there are not more than two components in the mixure. Accordingly, the same approach as earlier can be carried forward (using K=2), with the caveat that we now have two mean function estimates $\hat{\mu}^1(t)$, $\hat{\mu}^2(t)$ (resp. $\hat{\rho}^1(t)$, $\hat{\rho}^2(t)$). The particular sensors that exhibit this behavior can be ascertained beforehand, and this multi-modal analysis can be done only in such cases.

B. Generative Modeling: Synthetic Data Generation

Note that the above approach also yields a generative model for information-bearning sensors as well. Indeed, in the standard diffusive SDE case, assuming that W(t) is a Brownian motion, we know from standard SDE theory that the density p(x,t) of S(t) is a solution of the Fokker-Planck

$$\frac{\partial}{\partial t}p(x,t) = -\frac{\partial}{\partial x}\mu(x,t)p(x,t) + \frac{\partial^2}{\partial x^2}\rho(x,t)p(x,t). \quad \text{(5)} \quad \text{where $\bar{a}(t) = \mathbb{E}[a(t)]$, $R(t) = \mathbb{E}[S(t)^2]$.}$$



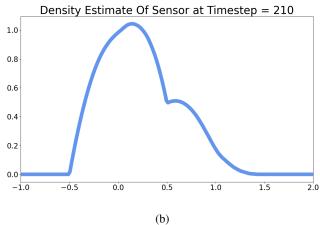


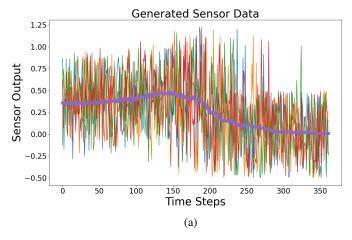
Fig. 2: (Evolution of the histogram over time for the same sensor. Notice the shift from unimodal to bimodal behavior.

In our current case, we do not have such a clear mathematical form for the evolution of the density. However, we can sample from the K-means density estimate that we used to compute the mean process at each time instant $t_k \in \Delta$ to obtain generated sample paths. This is a unique feature of our approach, and yields a physics-informed synthetic data generation mechanism that can be used to augment the raw (real) sensor data. Figures 3a and 3b show how we can generate samples from the time-varying densities which may possibly be multi-modal. The samples that are generated evidently have similar mean/variance properties as the "real" data. Notice that, since dW(t) is assumed to be a zero-mean process, we trivially have

$$\frac{d\mu(t)}{dt} = \bar{a}(t),\tag{6}$$

$$\frac{d\mu(t)}{dt} = \bar{a}(t),$$

$$\frac{d\rho(t)}{dt} = \frac{dR(t)}{dt} - 2\mu(t)\bar{a}(t),$$
(6)



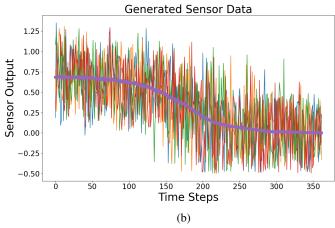


Fig. 3: Sample generated (synthetic) data from the estimated distributions. In bold is the mean of the actual (real) data.

III. EXPERIMENTS

A. Dataset

We focus our experiments on the aforementioned C-MAPSS dataset produced by NASA, which is organized into four subsets corresponding to distinct aircraft operating conditions. Each subset contains synthetic sensor readings meant to simulate run-to-failure trajectories of turbofan engines. Each trajectory is usually a few hundred timesteps. We drop 8 of the 22 available sensors (sensors 1, 5, 6, 10, 16, 18, 19, and 22) and split each trajectory into 20-timestep windows, which are shuffled during training.

B. Deep Learning Model

Consistent with previous work [23], we use a small LSTM neural network to predict the RUL at the end of each 20-timestep window. Specifically, we use a single-layer LSTM with 12 hidden units and apply a two-layer MLP to the LSTM's hidden state after all 20 timesteps have been processed.

We train on each of the four operating conditions in isolation, which aligns with real-world applications. We train the

model to minimize mean squared error (MSE) of the RUL at the end of each 20-timestep window:

$$J = \frac{1}{B} \sum_{B} \left\| RUL_{t}^{pred} - RUL_{t}^{gt} \right\|^{2} \tag{8}$$

where B is the batch size (64 in our case), RUL_t^{pred} is the RUL predicted by the model at timestep t (end of the 20-timestep window), and RUL_t^{gt} is ground truth. During testing, we predict the RUL at the end of each trajectory in the test set and report MSE across the entire test set (same as Equation 8) along with the average L1 error. We report the average of these metrics across 3 random seeds.

We use an Adam optimizer and a learning rate of 0.001. We train the model for 100 epochs on the FD001 and FD003 subsets and for 1000 epochs on the FD002 and FD004 subsets. We also used early stopping, testing the best-performing model from any epoch.

C. Results

Table I presents the MSE and L1 error of the model under each operating condition. The leftmost columns indicate the operating condition and whether μ and/or ρ were included as input during each training run.

Under all four operating conditions, including μ and ρ significantly outperforms the models where neither μ or ρ are included. ρ , however, appears to contain less information than μ , as shown by our ρ -only ablations; these runs nonetheless outperformed the baseline under all operating conditions. FD002 and FD004 are clearly of much greater complexity than FD001 and FD003, which is consistent with previous work [23].

Condition	Mu	Rho	Test MSE	Test L1
FD001	X	X	4.73	1.21
FD001	/	X	1.24	0.55
FD001	X	1	2.13	0.91
FD001	1	✓	1.02	0.59
FD002	Х	Х	430.44	15.41
FD002	1	X	309.23	13.15
FD002	X	1	282.44	12.62
FD002	1	✓	277.66	12.52
FD003	Х	Х	3.47	1.40
FD003	/	X	1.69	0.42
FD003	X	1	3.32	0.57
FD003	1	✓	0.15	0.29
FD004	Х	Х	944.21	22.50
FD004	1	X	670.47	19.05
FD004	X	/	720.76	19.57
FD004	/	✓	659.21	18.79

TABLE I: Test set mean squared error and L1 error under different operating conditions (FD001, FD002, FD003, FD004), according to whether mu and rho were added as inputs. Results are averaged across 3 seeds.

IV. SUMMARY AND CONCLUSIONS

We have presented a physics informed machine learning (PIML) approach to solving the problem of remaining useful lifetime (RUL) prediction using the C-MAPSS dataset. Our approach, based on stochastic models of the underlying data, is

novel in that we do not use the traditional PIML architectures such as PINNs, but rather augment the training data using estimated underlying quantities. Finally, our apprach was shown to allow for generative synthesis of data (i.e. physics aware synthetic data generation). The statistical distribution of the sensor data was seen to exhibit multi-modal behavior, and our approach is valid in this situation as well. Our experiments clearly indicate the increased accuracy (measured both using MSE and absolute error) over traditional deep learning (LSTM) approaches. It is expected that future work in this area would involve increasing the scale and scope of physics informed ML methods for predictive diagnostics.

REFERENCES

- Y. Huan, Q. Hu, Y. Jin, and G. Hu, "Fault detection and prognostics for avionics systems: A review," *Aerospace Science and Technology*, vol. 99, p. 106012, 2020.
- [2] S. Singh, K. Iqbal, and R. Pathak, "A review on machine learning applications in aircraft maintenance," *Aerospace Science and Technology*, vol. 101, p. 106052, 2020.
- [3] Z. Tian, S.-C. Tso, Y. Li, D. Xie, and K.-L. Tsui, "Health monitoring systems and prognostics for aircraft structures: A review," *Aerospace Science and Technology*, vol. 101, p. 106038, 2020.
- [4] X. Tong, Y. Zhang, and X. Chen, "A review of prognostics and health management for aircraft systems," *Aerospace Science and Technology*, vol. 78, pp. 469–476, 2018.
- [5] Z. Wang, D. Zhou, H. Jiang, and X. Shao, "Data-driven prognostics for aircraft components: A review," *Aerospace Science and Technology*, vol. 89, pp. 654–662, 2019.
- [6] D. Yang, Y. Xu, and D. Wang, "Condition-based maintenance for aircraft engines based on big data analytics," *Aerospace Science and Technology*, vol. 94, pp. 74–82, 2019.

[7]

- [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," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
- [9] J. Xu, H. Xiao, and G. E. Karniadakis, "Physics-informed neural networks: A comprehensive review," *Journal of Computational Physics*, vol. 404, p. 109180, 2020.
- [10] W. Zhang, Z. Qi, J. Chen, L. Wang, and L. Tang, "Physics-informed machine learning: A survey on physics-constrained neural networks," arXiv preprint arXiv:2003.10811, 2020.
- [11] Q. Yang, H. J. Kang, Y. Zhang, and D. N. Metaxas, "Physics-informed neural networks for inverse problems in imaging: A review," *Computer Methods in Applied Mechanics and Engineering*, vol. 372, p. 113391, 2020.
- [12] A. Ghavamian, E. Arabi, M. Gholami, and A. Borji, "Physics-informed machine learning for augmented reality flight simulation," *Aerospace Science and Technology*, vol. 101, p. 106042, 2020.
- [13] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016.
- [14] J. Wang and P. Perdikaris, "Physics-informed deep learning for inverse problems in fluid dynamics," arXiv preprint arXiv:2001.04271, 2020.
- [15] W. Chen, L. Zhang, and D. Lee, "Physics-informed machine learning for material science," *Advanced Materials*, vol. 25, no. 4, pp. 512–525, 2023.
- [16] Y. Zhu, J. Qiao, and N. Zabaras, "Physics-informed deep learning for structural health monitoring and non-destructive evaluation: A review," *Mechanical Systems and Signal Processing*, vol. 148, p. 107194, 2021.
- [17] J. Hamilton, I. Frosio, S. Tyree, and M. Prasad, "Learning physics from medical imaging using self-supervised deep neural networks," *Nature Machine Intelligence*, vol. 3, pp. 780–791, 2021.
- [18] J. Onieva, F. Santini, S. Nikolenko, and S. Bakas, "Physics-informed deep learning for dynamic contrast-enhanced mri reconstruction," arXiv preprint arXiv:2002.05700, 2020.

- [19] Y. Zhu and N. Zabaras, "Physics-informed neural networks for high-dimensional surrogate modeling and uncertainty quantification without labeled data," *Journal of Computational Physics*, vol. 403, p. 109151, 2020.
- [20] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," arXiv preprint arXiv:1910.07291, 2019.
- [21] A. Saxena, J. R. Celaya, B. Saha, and K. Goebel, "Damage propagation modeling for aircraft engine run-to-failure simulation," *Aerospace Science and Technology*, vol. 12, no. 4, pp. 277–288, 2008.
- [22] X.-S. Si, W. Wang, and C. Hu, "Prognostics of equipment health: A review of data-driven methods and hybrid approaches with implementation guidelines," in 2011 IEEE International Conference on Prognostics and Health Management. IEEE, 2011, pp. 1–10.
- [23] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), 2017, pp. 88–95.
- [24] X. Jiang, Y. Yang, F. Dong, H. Jiang, and Y. Miao, "Hybrid lstm-rnn model with adversarial training for remaining useful life prediction," *Journal of Mechanical Science and Technology*, vol. 34, no. 4, pp. 1491– 1502, 2020.
- [25] Y. Yang, F. Dong, Y. Miao, J. Yan, and J. Lee, "Deep learning approach for remaining useful life prediction based on lstm networks," in 2018 Prognostics and System Health Management Conference (PHM-Asia). IEEE, 2018, pp. 1–6.
- [26] K. Zhou, Z. Zhao, K. Mao, B. Hu, and Y. Zhang, "Remaining useful life estimation for turbofan engines based on a convolutional long shortterm memory recurrent neural network," *Journal of Mechanical Science* and Technology, vol. 32, no. 9, pp. 4287–4297, 2018.