

Data-Driven Prognostics with Multi-Layer Perceptron Particle Filter: a Cross-Industry Exploration

Francesco Cancelliere¹, Sylvain Girard², Jean-Marc Bourinet³

^{1,2} *PHIMECA, Paris, 75012, France*

cancelliere@phimeca.com

girard@phimeca.com

^{1,3} *SIGMA Clermont University, Aubiere, 63178, France*

jean-marc.bourinet@sigma-clermont.fr

ABSTRACT

The integration of particle or Kalman filters with machine learning tools like support vector machines, Gaussian processes, or neural networks has seen extensive exploration in the context of prognostic and health management, particularly in model-based applications. This paper focuses on the Multi-Layer Perceptron Particle Filter (MLP-PF), a data-driven approach that harnesses the non-linearity of MLP to describe degradation trajectories without relying on a physical model. The Bayesian nature of the particle filter is utilized to update MLP parameters, providing flexibility to the method and accommodating unexpected changes in the degradation behavior.

To showcase the versatility of MLP-PF, this work demonstrates its seamless integration into diverse use cases, such as lithium-ion battery analysis, virtual health monitoring for turbofans, and the assessment of fatigue crack growth. We illustrate how it effortlessly accommodates various contexts through slight parameter modifications. Adjustment includes variation in the number of neurons or layers in the MLP, threshold adjustments, initial training refinements and the adaptation of the process noise. Addressing different degradation processes across these applications, MLP-PF proves its adaptability and utility in various contexts.

These findings highlight the method's versatility in adapting to diverse use cases and its potential as a robust prognostic tool across various industries. MLP-PF offers a practical and efficient means of estimating remaining useful life and predicting degradation in complex systems, with implications for advancing prognostic tools in diverse applications.

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

1. INTRODUCTION

Prognostic and Health Management (PHM) plays a crucial role in engineering by aiming to estimate the health state, detect early failures, and predict the remaining useful life of systems or components (Zio, 2022). Implementing PHM algorithms allows for condition-based or predictive maintenance strategies, ultimately optimizing maintenance frequency and reducing operational costs (Bailey, Sutharssan, Yin, & Stoyanov, 2015). Traditional physics-based methods in PHM rely on known equations, and demand extensive domain knowledge while generally being computationally expensive, limiting their real-time applicability (Chang, Fang, & Zhang, 2017).

In contrast, the rise of data availability in recent years coincides with the exploration of data-driven methods such as neural networks, random forests, and support vector machines (Wang, Jin, Deng, & Fernandez, 2021; Hu, Xu, Lin, & Pecht, 2020; Vanem et al., 2023). However, these methods often face challenges related to data quantity, quality, and generalization across unseen conditions. To overcome these hurdles, hybrid approaches have been proposed (Cancelliere, Girard, Bourinet, & Broggi, 2023; Li et al., 2024), aiming to combine the strengths of data-driven and physics-based methods.

Among the hybrid approaches, a common one consists of integrating particle or Kalman filters with machine learning tools like neural networks or support vector machines (Dong, Jin, Lou, & Wang, 2014; Jha, Bressel, Ould-Bouamama, & Dauphin-Tanguy, 2016). In these frameworks, machine learning tools act as surrogates for physics-based models, reducing the need for extensive domain expertise. Meanwhile, Bayesian filters allow to quantify the uncertainties associated with the prediction, enhancing the robustness of the approach. Other works, such as (Ma, Karkus, Hsu, & Lee, 2020) or (Ge, Sun, & Ma, 2019) proposed combination of PF with, respectively, recurrent neural network (RNN) and long-short term memory

network (LSTM). A more comprehensive review of combination of PF and data driven techniques can be found in (Reza et al., 2024).

A combination of radial basis functions with particle filters, initially proposed by (Sbarufatti, Corbetta, Giglio, & Cadini, 2018), presented a novel approach to estimate the state of charge of lithium-ion batteries, which was later extended to predict the end of life for batteries by replacing the surrogate model with a Multi-Layer Perceptron (MLP) neural network (Cadini, Sbarufatti, Cancelliere, & Giglio, 2019). This adaptation, called Multi-Layer Perceptron Particle Filter (MLP-PF), capitalizes on the non-linear nature of MLPs to describe system degradation trajectories and leverages the Bayesian framework of particle filters to adjust to incoming measurements.

The primary contribution of this work lies in the application of the MLP-PF to three different case studies. By demonstrating the versatility of MLP-PF, this study showcases its seamless integration into diverse applications, starting from the case of lithium-ion batteries, then changing to the estimation of a virtual health indicator for turbofans, and the assessment of fatigue crack growth. Through minor parameter modifications such as variations in MLP architecture, threshold adjustments, initial training refinements, and adaptation of process noise levels, MLP-PF effortlessly accommodates various contexts.

Differing from conventional data-driven approaches, this study adopts a single historical degradation trajectory as training for the MLP neural network. The training serves merely as a starting point for the (PF) to explore the state-space, relying on its Bayesian nature to discern the hidden degradation dynamics. This approach furnish the algorithm with exceptional adaptability while significantly mitigating the need for extensive historical data, a primary drawback of traditional data-driven methods.

To evaluate the algorithm’s performance, various metrics including Relative Accuracy, Confidence Interval Coverage (Jules, Cancelliere, Mattrand, & Bourinet, 2023), and the β Metric (Lall, Lowe, & Goebel, 2013) are employed. These metrics assess not only accuracy and precision but also consider the uncertainty associated with the predictions, providing a comprehensive evaluation framework.

Addressing various degradation processes across different applications highlights the adaptability and utility of the MLP-PF. These findings emphasize the method’s versatility in accommodating diverse use cases and underscore its potential as a robust prognostic tool across multiple industries. MLP-PF provides a practical and efficient means of estimating remaining useful life and predicting degradation in complex systems, thereby advancing prognostic tools across a broad spectrum of applications.

The structure of this paper is the following: Section 2 briefly describes the proposed method and the metrics used to evaluate performance. Following this, Section 3 introduces the three use cases addressed in this study, while Section 3.1, Section 3.2 and Section 3.3 present the results corresponding to each case. Finally, Section 4 draws conclusions and provides perspectives on this work.

2. MULTI LAYER PERCEPTRON PARTICLE FILTER

The method employed in this work was first proposed by (Sbarufatti et al., 2018), where a combination of radial basis function neural networks and particle filters was used to estimate the state of charge of lithium-ion batteries. The method was later improved by (Cadini et al., 2019), where it was extended to estimate the state of health of the battery. In this work, the method is applied to three different use cases to showcase its ability to adapt to different contexts with slight changes in the hyperparameters.

The multi-layer perceptron neural network is used as a surrogate for the given degradation model, such as the turbofan VHI or the batteries’ capacity. In all cases, it consists of a single input, which is the discrete time step k , and a single output \tilde{g} , representing the predicted value at the given time step. The decision to use a simple neural network, such as an MLP, is driven by the necessity for flexibility and the desire to minimize the number of parameters estimated by the PF. Despite its simplicity, an MLP remains capable of capturing the nonlinearities inherent in the data. This choice strikes a balance between model complexity and computational efficiency, enabling effective integration with the PF framework.

The internal architecture of the network (number of layers, number of neurons per layer, and the activation functions) is case-dependent, particularly on the shape of the degradation trajectories and to ensure computational times are compatible with the given context (higher the network complexity, higher the computational time). The parameters of the network, meaning its weights and biases, are then packed into a vector x_k . The starting parameters x_0 are obtained by training the network based on a known run-to-failure degradation process, as can be observed in the (a) figures of the three cases.

The particle filter (Arulampalam, Maskell, Gordon, & Clapp, 2002) is a sequential Monte-Carlo algorithm that generates a set of particles which are used to estimate the posterior probability density function (PDF) of a hidden state, which in our case are the parameters x_k . Hence, a set of N_s copies (i.e particles) of x_0 is generated based on:

$$x_k^i = x_{k-1}^i + \omega_{k-1} \quad (1)$$

where i is the index of the particles and ω is the process noise, which is an hyperparameter that has to be carefully tuned. Each x_k^i contains the parameters of a MLP, which,

when propagated through the network, generates a prediction of a possible degradation trajectory. The PF operates by applying a prediction-update recurrence. The predictions at time step $k - 1$ serve as the prior PDF, which is updated at each subsequent time step k upon the arrival of new observations. The update of the particles is performed by computing their likelihood, which indicates how close the i^{th} degradation trajectory is to the actual observations. The likelihood \mathcal{L}_k^i of each particle i is computed as:

$$\mathcal{L}_k^i = p(z_{0:k}|x_k^i) = ((2\pi)^{k+1}|\Sigma_\eta|)^{-0.5} \exp \left\{ -\frac{1}{2} (z_{0:k} - \tilde{g}(x_k^i, 0 : k))^T \Sigma_\eta^{-1} (z_{0:k} - \tilde{g}(x_k^i, 0 : k)) \right\} \quad (2)$$

where $z_{0:k}$ are the observations from 0 to k , $\tilde{g}(x_k^i, 0 : k)$ is the prediction of the network for the i^{th} particle and Σ_η is the diagonal covariance matrix, with diagonal element equal to η , representing the measurement noise and assumed Gaussian. \mathcal{L}_k^i is the probability of obtaining the measurement z_k given the i^{th} prediction \tilde{g}_k , and it is used as importance weight w_k^i for the particles.

To finally construct the posterior pdf the sampling importance resampling (SIR) algorithm is employed (Doucet, Godsill, & Andrieu, 2000): the weights are normalized, and the particles are resampled based on the normalized weights \tilde{w}_k . The closer the prediction is to the measurements, the higher the normalized importance weight of that particle, meaning that the particle is more likely to be resampled, which signifies that it is closer to the actual degradation trajectory of the observed process.

The importance weights of the particles are also utilized to enforce specific conditions, ensuring that particles adhere to desired behaviors. One example is imposing the monotonicity of the trajectory or setting bounds on the output value (e.g., ensuring it is always greater than 0). If a particle violates the specified condition, its weight is set to zero, indicating that it will not be resampled. Instead, it is replaced by a particle with a higher importance weight, thereby maintaining compliance to the desired conditions.

The collection of normalized particles at time step k , each representing a potential degradation trajectory, enables the computation of the posterior probability density function of the degradation state in future time steps. Consequently, it becomes possible to calculate statistics related to predictions, such as the mean and relative uncertainties. Additionally, by establishing a threshold for the end of life of the system, it becomes feasible to determine the distribution of the End of Life $p(\text{EOL}_k|z_{0:k})$, and consequently the RUL_k as:

$$\text{RUL}_k = \text{EOL}_k - k \quad (3)$$

To evaluate the performance of the algorithm we use three

different metrics. The first one is the cumulative relative accuracy, defined as:

$$\text{CRA} = \frac{1}{T_{fail}} \sum_{k=0}^{T_{fail}} \left(1 - \left| \frac{\text{RUL}_k^{\text{actual}} - \text{RUL}_k^{\text{pred}}}{\text{RUL}_k^{\text{actual}}} \right| \right) \quad (4)$$

where T_{fail} is the time step at which the system fails. This represents the distance of the prediction to the actual EOL, evaluated at each time step. A perfect prediction has a value of 1.

The second metric is the confidence interval coverage (Jules, Cancelliere, et al., 2023), which is used to assess the prediction considering the confidence interval, and is defined as:

$$\text{CIC} = \frac{1}{T_{fail}} \sum_{k=0}^{T_{fail}} \mathbb{1}_{\text{RUL}_k^{\text{actual}} \in \widehat{\text{CI}}_k} \quad (5)$$

where $\mathbb{1}_{\text{RUL}_k^{\text{actual}} \in \widehat{\text{CI}}_k}$ is the indicator function that takes one if the actual RUL lies in the predicted confidence interval, 0 otherwise. If the prediction at each time step include the $\text{RUL}^{\text{actual}}$, the CIC is going to be 1, while 0 if the true RUL is always outside the confidence interval.

The last indicator is called the β metric (Lall et al., 2013), which represent the area of the predictions that falls inside the α bound.

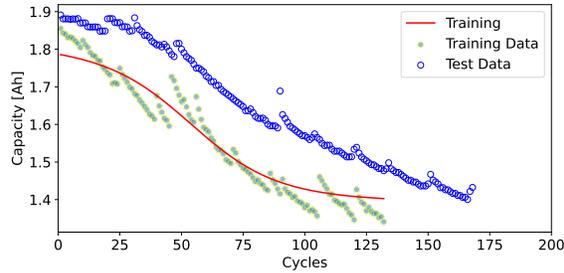
$$\beta_k = \frac{1}{T_{fail}} \sum_{k=0}^{T_{fail}} \int_{\text{RUL}_k - \alpha}^{\text{RUL}_k + \alpha} \text{PDF}(\text{RUL}) d\text{RUL} \quad (6)$$

The α bounds are defined as $\text{RUL}^{\text{actual}} \pm \alpha$. The β metric evaluates the accuracy of predicted RUL bounds compared to true RUL bounds, considering a specified uncertainty level (α). It quantifies the overlap between predicted and true RUL bounds normalized by the true RUL length. Higher values indicate better agreement between predicted and true bounds, reflecting improved prediction accuracy.

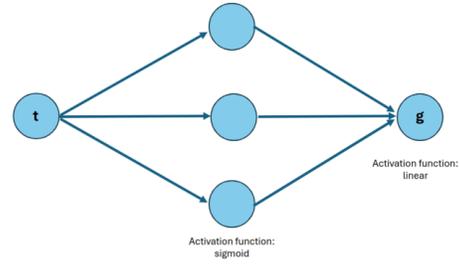
3. USE CASES

The proposed approach will now be applied to three use cases: estimating the end of life of lithium-ion batteries based on their decreasing capacity, propagating a virtual health indicator developed to estimate the state of health of turbfans, and modeling the growth of a fatigue crack in a panel. Although these cases share a time-dependency, their degradation processes differ significantly in terms of shape and rapidity. Therefore, we employ three different MLP network architectures, each tailored to the specific characteristics of its respective case.

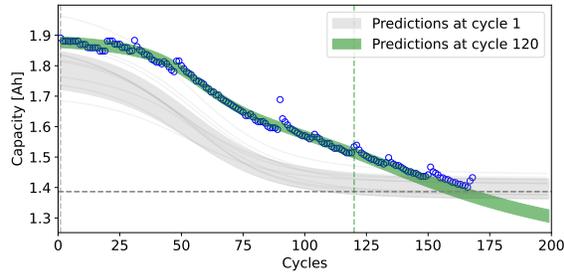
For consistency and comparison, in each cases we use the same number of particles, $N_s = 1000$, and the same number of epochs for the initial training (epochs = 500). Additionally, we employ a decreasing variance, defined as:



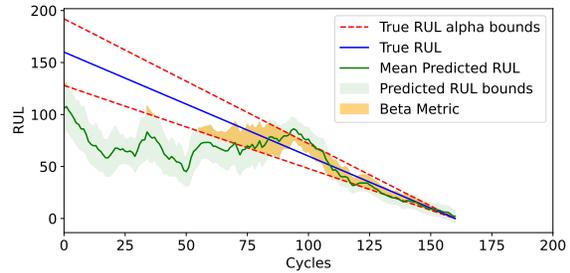
(a) Training and testing dataset for Li-Ion batteries. The red line is the output of the trained MLP.



(b) Architecture of the MLP neural network used for the Li-Ion batteries case.



(c) Two instant of time of the simulation. The grey dotted line is the end of life threshold.



(d) Remaining Useful Life prediction in terms of mean and uncertainty bounds.

Figure 1. Lithium-Ion Battery case.

$$\omega_k = \sigma_0 e^{-\frac{k}{\sigma_1}} + \sigma_2 \quad (7)$$

Here, ω perturbs the MLP parameters, as described in Eq. (1).

The use of a decreasing process variance is crucial for the convergence of the algorithm. Initially, a higher variance (which practically signifies a higher perturbation of x_k in Eq. (1)) is necessary to explore the state space and adapt to the first incoming observations, especially if these are significantly different from the training data. As more observations become available, the variance is reduced to reflect the increased information about the actual system. This reduction in variance helps prevent a single observation, especially a noisy one, from excessively perturbing the prediction. This strategy ensures a balanced adaptation process, enabling the algorithm to remain robust against noisy observations while gradually refining its predictions.

Given that the complexity of the network correlates with the number of parameters, using the same ω value for different architectures will lead to different perturbation. Specifically, higher complexity requires lower perturbation (i.e. lower ω) to prevent degeneration. If the MLP parameters change too rapidly, they may lose meaning and connection with prior information. Hence, the σ parameters and the measurement noise η , responsible for computing the particle likelihood in Eq. (2), vary across different cases.

Furthermore, in all three cases, we opt to use a single trajec-

tory for the initial training. This choice aims to demonstrate the algorithm’s ability to adapt to varying conditions and its capacity to achieve satisfactory performance in predicting the RUL without requiring a large amount of data. The proposed metrics are evaluated throughout all the degradation process and in the last 25% of life. This evaluation demonstrates that the algorithm’s performance improves over time as more information becomes available, and it converges to the target data even when the initial training data differ significantly.

3.1. Lithium-ion Batteries

The dataset used for the first use case is the one developed by NASA for the prognostic and diagnostic analysis of batteries (Saha & Goebel, 2007). The capacity of batteries decreases over time due to usage and electrochemical reactions occurring inside the battery. The end of life of batteries is typically defined when the capacity drops below 80% of the initial capacity. However, to make the most of the dataset, in this work, we set a threshold of 1% higher than the last point, which is 1.42 Ah. In Fig. 1(a) the two batteries used for the initial training of the network (battery 18 of the dataset) and for testing (battery 7) are shown.

The architecture of the network consists of a single hidden layer with 3 neurons, where the activation functions are a sigmoid for the hidden layer and linear for the output layer. The network structure is depicted in Fig. 1(b). This results in a total of 6 weights and 4 biases, which, after training, are stacked

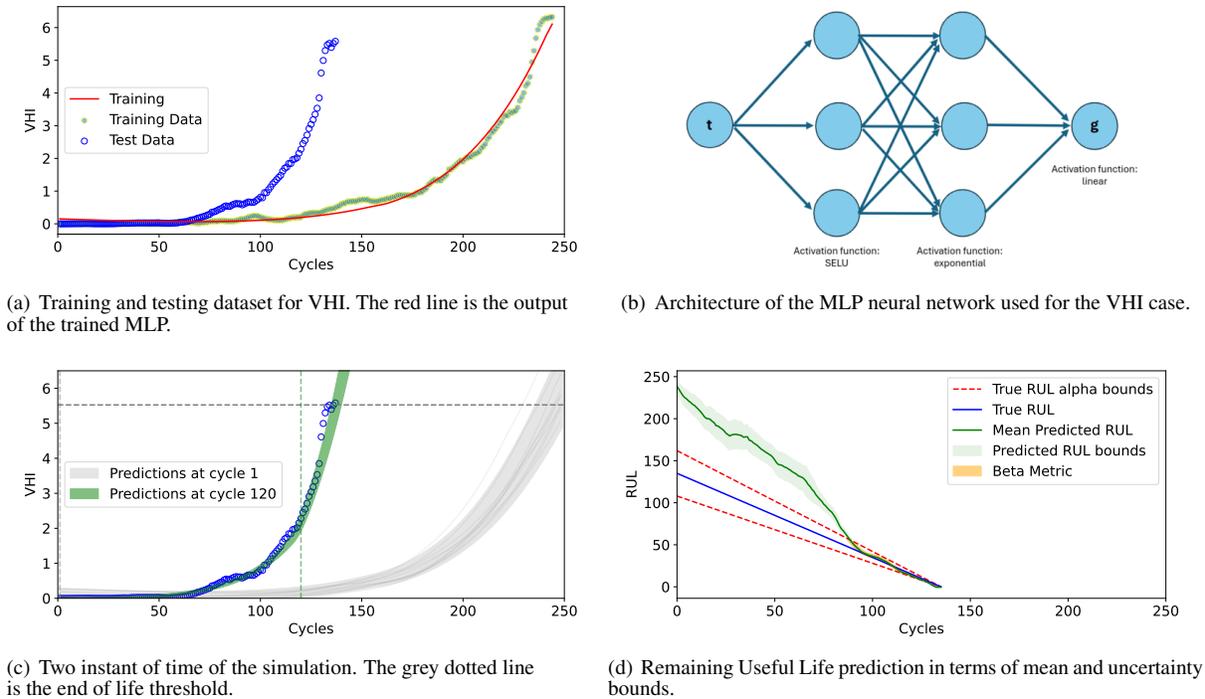


Figure 2. Virtual Health Indicator case.

in the vector x_0 , resulting in 10 parameters. The output of the trained network with these parameters is represented by the red line in Fig. 1(a).

The initial process noise is taken as $\sigma_0 = 5 \times 10^{-2}$, while the floor value is set to $\sigma_2 = 10^{-5}$, with a decreasing rate σ_1 of 50. The measurement noise is set to 10^{-2} . Due to the simplicity of the network, the initial value σ_0 is relatively high, providing more flexibility to the algorithm. Furthermore, the adaptability of the algorithm is necessary due to the intrinsic nature of lithium-ion batteries, which can perform differently from one another, as observed in Fig. 1(a). The initial perturbation, obtained by applying Eq. (1) to each of the N_s particle can be observed in Fig. 1(c) as the grey lines.

The results of the simulation are presented in Fig. 1(c) and Fig. 1(d). In the first, two instances of time, at the beginning and about the end of the simulation, are shown, highlighting the adaptability of the algorithm. Starting from the initial training, the algorithm adapts to incoming measurements and estimates the new degradation behavior. The last figure shows the results in terms of remaining useful life estimation. Initially, the predictions were more related to the training data, which has a faster end of life, while converging to the actual RUL at about the halfway point of the battery’s lifetime. The evaluation of the algorithm’s performance is reported in Table 1, where it can be observed that all the metrics improved when evaluated in the last 25% of the lifetime. Particularly, the Confidence Interval Coverage 25 has a value of 1, indicat-

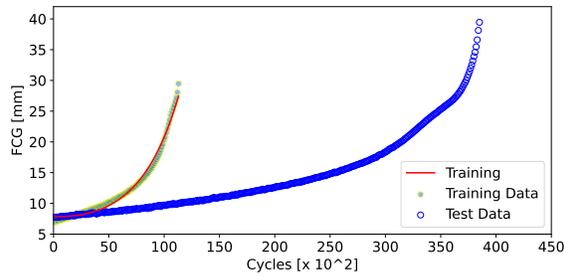
ing that the actual RUL has always been inside the predicted bounds.

3.2. Virtual Health Indicator

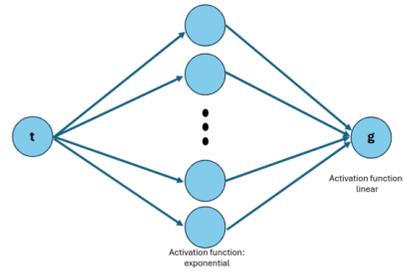
The second use case proposed here involves the estimation of the future behavior of a virtual health indicator developed for estimating the state of health of turbofans (Jules, Mattrand, & Bourinet, 2023). This VHI measures the degradation of turbofans, thus, opposite to the batteries case, it exhibits an upward trajectory, where a higher value indicates higher degradation. Similarly to the previous case, the end-of-life threshold has been set to utilize the maximum available number of cycles from the test dataset.

Upon observing the historical data of the VHI, it can be noted that initially, it exhibits a flat trajectory, remaining nearly at zero until the degradation process begins, after which it adopts an exponential-like trajectory. To accommodate this, the proposed network for this case consists of two hidden layers with 3 neurons each. The first layer employs a scaled exponential linear unit (SELU) activation function, while the second layer employs an exponential activation function. The output layer uses a linear activation function. The structure of this network (see Fig. 2(b)) is thus more complex, consisting of a total of 22 parameters (15 weights and 7 biases).

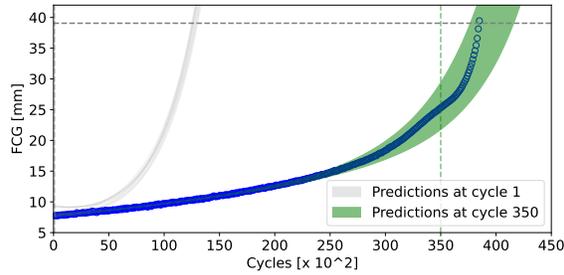
As mentioned in Section 2, certain conditions can be enforced on the particles to help them meet specific constraints. In this case, since the VHI has been designed to be greater than 0



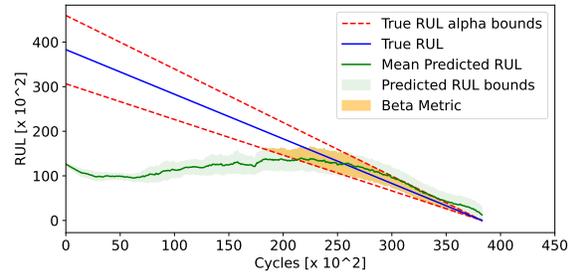
(a) Training and testing dataset for FCG. The red line is the output of the trained MLP.



(b) Architecture of the MLP neural network used for the FCG case with 20 neurons in the hidden layer.



(c) Two instant of time of the simulation. The grey dotted line is the end of life threshold.



(d) Remaining Useful Life prediction in terms of mean and uncertainty bounds.

Figure 3. Fatigue Crack Growth case.

and monotonous, these two conditions have been enforced by eliminating particles at each iteration that did not adhere to them. With more parameters, the initial process noise had to be slightly reduced, and particularly, we set the value of $\sigma_0 = 10^{-2}$. The other values (floor noise σ_2 and the measurement noise η) remained unchanged.

Fig. 2(a) displays the training and test data, as well as the output of the trained network. It can be observed that the shape of the two trajectories is similar; however, the rate of degradation varies notably, with the training dataset exhibiting a slower trend. Additionally, the initial flat plateau adds complexity to the prediction task since the algorithm receives measurements close to the expected values, resulting in high likelihood. This behavior is illustrated in Fig. 2(d), where initially, the predicted RUL decreases almost constantly, indicating little variation in prediction. Once the measurements from the VHI start to increase, signaling the onset of degradation, the algorithm quickly adapts to the new degradation behavior and converges to the actual RUL.

Even in this case, all metrics improved when evaluated in the last quarter of the lifetime. We note a relatively low CIC in this case, which can be attributed to the narrow prediction bounds. On the other hand, this led to a relatively high value of the β Metric of the last quarter.

3.3. Fatigue Crack Growth

The last use case concerns the propagation of a crack in a rectangular plate of commercial 316L steel (Langlois Raphael, 2018) subjected to a fatigue load. The dataset comprises two tests of identical plates, with a cycling tensile loading applied with a frequency of 10 Hz and a R ratio of 0.1. The first one is subjected to a maximum force of 15 kN (test data, see Fig. 3(a)) and the second to a maximum force of 22.5 kN (training data, also in Fig. 3(a)), with a As expected, the higher the applied force, the faster the crack propagates.

The trajectories follow an exponential-like function. Therefore, the proposed architecture for this problem consists of a single hidden layer with an exponential activation function (see Fig. 3(b)). To enforce the exponential behavior of the MLP (and also to challenge the algorithm), the hidden layer consists of 20 neurons, nearly tripling the number of parameters to 61.

Due to the higher number of parameters, the values of σ_0 and σ_2 have to be significantly decreased. For this simulation, they have been set to $\sigma_0 = 5 \times 10^{-4}$ and $\sigma_2 = 10^{-6}$. As in the VHI case, the monotonicity of the curve is enforced, particularly since this is a physical constraint.

In contrast to the VHI case, we use the faster degradation as training data while attempting to estimate the slower trajectory. This poses a challenge for the algorithm since extrapolating future data without prior examples is inherently

Table 1. RUL Evaluation Metrics.

	Li-Ion	VHI	FCG
CRA	0.703	0.474	0.489
CRA 25	0.772	0.762	0.240
CIC	0.475	0.140	0.425
CIC 25	1.000	0.294	1.000
β	0.326	0.277	0.273
β 25	0.465	0.737	0.404

complex for data-driven algorithms. Nonetheless, even in this case, we observe that the algorithm adapts quite rapidly, with the remaining useful life initially remaining constant (see Fig. 3(d)), indicating that the algorithm recognized early on that the training degradation was faster. As the correct RUL is approached around the halfway point of the lifetime, the algorithm is able to capture the new trajectory and remains consistent with the prediction.

In terms of metric, we note that the CRA is low, especially in the last 25%. This is mainly due to the relative error in the very last points, where even a small error in the average predicted RUL leads to a significant penalization of the CRA, as the actual RUL is a small number. In contrast, we observe a perfect coverage in the last quarter, as the actual RUL has always fallen within the predicted bounds during that period.

4. CONCLUSIONS

The proposed methodology, combining multi-layer perceptron neural networks and particle filters, demonstrated its adaptability and effectiveness in estimating the remaining useful life across diverse engineering systems. By applying the method to three distinct use cases – estimating the end of life of lithium-ion batteries, predicting the behavior of a Virtual Health Indicator in turbofans, and analyzing the propagation of fatigue cracks in steel plates – we showcased its versatility and accuracy in capturing degradation processes. The utilization of a single training history for each case underscores the robustness of the algorithm and its adaptability even when limited data about the target system are available. Evaluation metrics such as Cumulative Relative Accuracy (CRA), Confidence Interval Coverage (CIC), and the β metric provided valuable insights into the accuracy, coverage, and uncertainty of predicted RUL bounds. These findings emphasize the practical implications of accurate RUL estimation in predictive maintenance, enabling proactive decision-making to optimize maintenance schedules and reduce operational costs. Further research can explore enhancements to the methodology and its application to additional use cases beyond time-dependent applications, thus enhancing its utility and effectiveness in real-world scenarios

ACKNOWLEDGMENT

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 955393

REFERENCES

- Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/nongaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174–188. doi: 10.1109/9780470544198.ch73
- Bailey, C., Sutharssan, T., Yin, C., & Stoyanov, S. (2015). Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms. *The Journal of Engineering*(July). doi: 10.1049/joe.2014.0303
- Cadini, F., Sbarufatti, C., Cancelliere, F., & Giglio, M. (2019). State-of-life prognosis and diagnosis of lithium-ion batteries by data-driven particle filters. *Applied Energy*, 235(June 2018), 661–672. doi: 10.1016/j.apenergy.2018.10.095
- Cancelliere, F., Girard, S., Bourinet, J.-M., & Broggi, M. (2023). Grey-box Approach for the Prognostic and Health Management of Lithium-Ion Batteries. *Annual Conference of the PHM Society*, 15(1), 1–8. doi: 10.36001/phmconf.2023.v15i1.3506
- Chang, Y., Fang, H., & Zhang, Y. (2017). A new hybrid method for the prediction of the remaining useful life of a lithium-ion battery. *Applied Energy*, 206, 1564–1578. doi: 10.1016/j.apenergy.2017.09.106
- Dong, H., Jin, X., Lou, Y., & Wang, C. (2014). Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter. *Journal of Power Sources*, 271, 114–123. doi: 10.1016/j.jpowsour.2014.07.176
- Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10, 197–208. doi: 10.1023/A:1008935410038
- Ge, Y., Sun, L., & Ma, J. (2019). An Improved PF Remaining Useful Life Prediction Method Based on Quantum Genetics and LSTM. *IEEE Access*, 7, 160241–160247. doi: 10.1109/ACCESS.2019.2951197
- Hu, X., Xu, L., Lin, X., & Pecht, M. (2020). Battery Lifetime Prognostics. *Joule*, 4(2), 310–346. doi: 10.1016/j.joule.2019.11.018
- Jha, M. S., Bressel, M., Ould-Bouamama, B., & Dauphin-Tanguy, G. (2016). Particle filter based hybrid prognostics of proton exchange membrane fuel cell in bond graph framework. *Computers and Chemical Engineering*, 95, 216–230. doi:

- 10.1016/j.compchemeng.2016.08.018
- Jules, E., Cancelliere, F., Mattrand, C., & Bourinet, J.-M. (2023). Remaining useful life prediction of turbofans with virtual health indicator: A comparison of particle filter-based approaches. , 75-82. doi: 10.1109/IC-SRS59833.2023.10381439
- Jules, E., Mattrand, C., & Bourinet, J.-M. (2023). Similarity learning for predictive maintenance: health indicator construction based on siamese neural networks and contrastive loss. *[Under Review]*.
- Lall, P., Lowe, R., & Goebel, K. (2013). Prognostic health monitoring for a micro-coil spring interconnect subjected to drop impacts. *PHM 2013 - 2013 IEEE International Conference on Prognostics and Health Management, Conference Proceedings*(June 2013). doi: 10.1109/ICPHM.2013.6621458
- Langlois Raphael, R. J., Coret Michel. (2018, November). Fatigue Crack Propagation Benchmark, GDR 3651 FATA CRACK. Retrieved from <https://doi.org/10.5281/zenodo.1478472> doi: 10.5281/zenodo.1478472
- Li, T., Chen, J., Yuan, S., Zarouchas, D., Sbarufatti, C., & Cadini, F. (2024). Particle filter-based fatigue damage prognosis by fusing multiple degradation models. *Structural Health Monitoring*, 0(0), 14759217231216697. doi: 10.1177/14759217231216697
- Ma, X., Karkus, P., Hsu, D., & Lee, W. S. (2020). Particle filter recurrent neural networks. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 5101–5108. doi: 10.1609/aaai.v34i04.5952
- Reza, M. S., Mannan, M., Mansor, M., Ker, P. J., Mahlia, T. M., & Hannan, M. A. (2024). Recent advancement of remaining useful life prediction of lithium-ion battery in electric vehicle applications: A review of modelling mechanisms, network configurations, factors, and outstanding issues. *Energy Reports*, 11(April), 4824–4848. doi: 10.1016/j.egy.2024.04.039
- Saha, B., & Goebel, K. (2007). *Battery Data Set. NASA Ames Prognostics Data Repository*.
- Sbarufatti, C., Corbetta, M., Giglio, M., & Cadini, F. (2018). Adaptive prognosis of lithium-ion batteries based on the combination of particle filters and radial basis function neural networks. *Journal of Power Sources*, 344, 128–140. doi: 10.1016/j.jpowsour.2017.01.105
- Vanem, E., Liang, Q., Ferreira, C., Agrell, C., Karandikar, N., Wang, S., ... Kandepu, R. (2023). Data-Driven Approaches to Diagnostics and State of Health Monitoring of Maritime Battery Systems. *Annual Conference of the PHM Society*, 15(1), 1–17. doi: 10.36001/phm-conf.2023.v15i1.3437
- Wang, S., Jin, S., Deng, D., & Fernandez, C. (2021). A Critical Review of Online Battery Remaining Useful Lifetime Prediction Methods. *Frontiers in Mechanical Engineering*, 7(August), 1–19. doi: 10.3389/fmech.2021.719718
- Zio, E. (2022). Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering and System Safety*, 218(PA), 108119. doi: 10.1016/j.ress.2021.108119