Conference papers   School of Computer Science

2023

Interpretable Input-Output Hidden Markov Model-Based Deep Reinforcement Learning for the Predictive Maintenance of Turbofan Engines

Ammar N. Abbas
Technological University Dublin, Ireland, ammar.abbas@tudublin.ie

Georgios C. Chasparis
Software Competence Center Hagenberg, Hagenberg, Austria

John Kelleher
Technological University Dublin, john.kelleher@tudublin.ie

Follow this and additional works at: https://arrow.tudublin.ie/scschcomcon

Part of the Computer Engineering Commons

Recommended Citation

Interpretable Input-Output Hidden Markov Model-Based Deep Reinforcement Learning for the Predictive Maintenance of Turbofan Engines ⋆

Ammar N. Abbas[1] , Georgios C. Chasparis[1] , and John D. Kelleher[2]

[1] Software Competence Center Hagenberg, Austria
{ammar.abbas, georgios.chasparis}@scch.at
[2] ADAPT Research Centre, Technological University of Dublin, Ireland
{john.d.kelleher}@tudublin.ie

Abstract. An open research question in deep reinforcement learning is how to focus the policy learning of key decisions within a sparse domain. This paper emphasizes on combining the advantages of input-output hidden Markov models and reinforcement learning. We propose a novel hierarchical modeling methodology that, at a high level, detects and interprets the root cause of a failure as well as the health degradation of the turbofan engine, while at a low level, provides the optimal replacement policy. This approach outperforms baseline deep reinforcement learning (DRL) models and has performance comparable to that of a state-of-the-art reinforcement learning system while being more interpretable.

Keywords: Deep Reinforcement Learning (DRL) · Input-Output Hidden Markov Model (IOHMM) · Predictive Maintenance · Interpretable AI

1 Introduction

Predictive maintenance can be categorized as (i) *Prognosis*: predicting failure and notifying for replacement or repair ahead of time (*Remaining Useful Life* or briefly RUL is usually used as a prognosis approach, which is the estimation of the remaining life of equipment or a system until it becomes non-functional [20]); (ii) *Diagnosis*: predicting the actual cause of failure in the future through cause-effect analysis, or (iii) *Proactive Maintenance*: anticipate and mitigate the failure modes and conditions before they develop [6]. While proactive maintenance captures the root cause of potential failure, predictive maintenance performs an overall data analytics to be able to ensure scheduled maintenance. In this paper, the aforementioned questions will be investigated in the context of predictive maintenance of turbofan engines [4,18].

Reinforcement Learning (RL) is a natural approach to solving time series-based stochastic decision problems, such as predictive maintenance [21], and has recently shown promising results. RL systems learn by interacting with the environment and can learn in an online setting without having the data set beforehand [22]. However, when the key policy decision learned by an RL agent is relatively rare in a data set (such as the decision of when to change the equipment before failure while maximizing its use), the policy can be dominated by irrelevant phenomena, resulting in inefficient training. At the same time, the derived optimal policy does not provide interpretations or the root cause of the failure, and therefore keeps humans out of the loop with limited collaborative intelligence. Furthermore, in real-world industrial environments, RL learns directly from the observed raw sensor data that does not provide information about the unobserved hidden factors responsible for the decision-making of the system such as its health, which can limit the RL agent to learning an optimal policy.

An Input-Output Hidden Markov Model (IOHMM) [2,17] is a form of Bayesian Network that involves probabilistic inference of latent variables. An IOHMM extends the standard HMM model by integrating the dependencies of various covariates (inputs) to the initial, transition, and emission probabilities [1]. It can overcome the challenges faced by RL through (i) learning unobserved states and interpretations based on those hidden states, (ii) combining multiple correlated sensor data, (iii) defining the state of the system and its hierarchical distribution based on its different levels of operation (normal, starting point of failure, close to failure, etc.), and (iv) dimensionality reduction based on the number of latent states that reduces the size and complexity of the raw data [24]. To address the need for a more direct and specialized data-based optimization, while maintaining the interpretability of the derived policies, we propose an unsupervised hierarchical modeling technique that combines a high-level IOHMM with a low-level Deep Reinforcement Learning (DRL) methodology for predictive maintenance.

Hierarchical modeling is a solution towards the sample-efficient RL, which decomposes the enormous long-horizon state space into several specialized short-horizon tasks. In the first step, the IOHMM prefilters large amounts of non-relevant data generated during the normal running of the equipment and detects the state at which failure is imminent. In the second step, the DRL agent learns a policy on equipment replacement conditioned on these (close to failure) states. Our experimental results indicate that the proposed state-/event-based approach with dynamic data pre-filtering has comparable *performance*[1] to prior work that trains RL agents directly on the full data set, hence increasing the training efficiency. Lastly, it allows for more explicit interpretability of the derived policies by learning the latent state space. Specifically, the IOHMM learns the hidden state representation of the system (x_t) and the DRL constructs the state-action pair modeling of the environment (s_t, a_t).

To evaluate our approach, we use the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), turbofan degradation data sets [4,18].

---

[1] performance indicates the ability to suggest replacement before failure with the use of the maximum usable life as well as with the least number of failed equipment

These data sets record the output from several engine units with multivariate time series sensor readings and operating conditions discretized based on the flight cycles within a run-to-failure simulation. The following subsets of these data sets will be used in this paper: **FD001** with 1 operating condition and 1 failure mode; **FD002** with 6 operating conditions and 1 failure mode; **FD003** with 1 operating condition and 2 failure modes; and, **DS01** with ground truth degradation values.

Structure: Section 2 provides the literature review. Section 3 frames predictive maintenance as an RL problem. Section 4 proposes the novel methodology. Section 5 explains the experimental setup. Section 6 provides the interpretability aspect of the proposed methodology. Finally, Section 7 compares the proposed architecture with the baseline and previous work.

2   Related Work

There have been several RL methodologies developed to optimize maintenance decisions. For this task, the effectiveness of an explainable adaptive event-driven RL strategy is shown in [13,15,16] where such agents can be deployed under situation-dependent adaptations. RL in industrial applications as a predictive maintenance strategy is shown in [11,14] where the model learns from both its own experience through environment interaction as well as from the human experience feedback. The work reported in [14,21] used turbofan engines [18] as their case study for optimal maintenance decisions and discussed the limitations of prior work. In particular, they highlight that prior work is often limited to estimating the RUL of a system, giving no cause-effect relationship between the failure and the components of the equipment.

In this paper, we take the *Bayesian particle filtering* approach (Monte Carlo simulation combined with DRL) proposed in [5] as the representative of the state-of-the-art DRL for industrial maintenance and use it as a benchmark for our work. In this benchmark methodology, sequential Monte Carlo simulation is used to map the raw sensor data into latent belief degradation states [21], and it is over these latent belief states (rather than the raw sensor data) that the deep reinforcement agent learns a policy for equipment maintenance.

Given the need for interpretable decisions, researchers have also investigated the use of the Hidden Markov Model (HMM) for predicting the RUL of turbofan engines. Recent research has demonstrated the effectiveness of HMMs both towards the interpretation of fault points in terms of a correlation between a sudden decrease in RUL and transition of HMM state, as well as in terms of predicting a failure event and degradation path [8,9]. In addition, the effectiveness of Input-Output HMMs (IOHMMs), which are a more generalized version of HMM, has been explored for the diagnosis of failure, prognosis, health status, and monitoring of RUL of industrial components [10,19]. The effectiveness of online HMM estimation-based Q learning that converges to a higher mean reward for the *Partially Observable Markov Decision Process (POMDP)*, where certain variables are hidden (not directly observable), is mathematically proven by [25].

Literature Gap and Research Contributions: The majority of the research on predictive maintenance using RL focuses on the prognosis based on the estimation of RUL from multivariate raw sensor readings. However, the interpretability of the faults of the machine (at the equipment level) is missing. Furthermore, realistic environments often have partial observability, where learning from raw data might lead to suboptimal decisions. Additionally, RL encounters learning inefficiency when trained with limited samples and in an online setting [7]. In this paper, a novel methodology is proposed for maintenance decisions and interpretability that is based on DRL. At a high level, an IOHMM is designed for detecting imminent-to-failure states, while at a low level, a DRL is designed for optimizing the optimal replacement policy. Furthermore, we present a comparative analysis with prior work that demonstrates the effectiveness of the proposed methodology in terms of both performance and interpretation.

3   Framing Predictive Maintenance as an RL Problem

In this section, the decision-making problem associated with optimal predictive maintenance is framed as an RL problem.

3.1   Environment Dynamics and Modeling

The DRL framework for predictive maintenance proposed in [14] considers three actions as a general methodology for any decision-making maintenance model; *hold* [2], *repair,* and *replace.* The constraints can be the maintenance budget, and the objective function can be the maximum uptime of the equipment. We propose a general framework for modeling such environments with state transitions based on the actions selected under stochastic events (uncertainty of failure, and randomness of replacement by new equipment) at any state, as illustrated in Figure 1. Although the general framework presented in Figure 1 includes three actions (hold, replace, and repair), the data sets used in the experiments reported in this paper do not include data on repair actions and so for these experiments, the action space consists of just two actions (hold or replace).
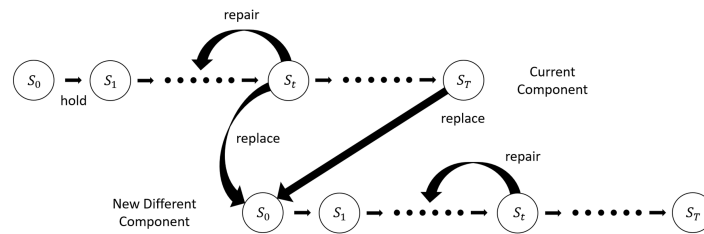

Fig. 1. Dynamics of the model of the environment

_____

[2] The action of hold means that the agent neither suggests to replace nor repair and the system is healthy enough for the next operating cycle.

3.2 Reward Formulation

For the maintenance decision having only replacement or hold actions, a dynamic reward structure has been formulated as shown in Equation (1) from [21]. In this equation c_r is the replacement cost, c_f is the failure cost, t is the current cycle, T_j is the final (failure) cycle, and r_t is the immediate reward.

$$r_t = \begin{cases} 0, & a_t = \text{Hold} & \& \ t < T_j, \\ -\frac{c_r}{t}, & a_t = \text{Replace} & \& \ t < T_j, \\ -\frac{c_r+c_f}{T_j}, & a_t = \text{Hold} & \& \ t = T_j, \\ -\frac{c_r+c_f}{T_j}, & a_t = \text{Replace} & \& \ t = T_j. \end{cases} \tag{1}$$

3.3 Evaluation Criteria

To evaluate the performance of the RL agent, these numerical values were chosen:

Cost The average optimal total return $(\widetilde{Q^*})$ serves as a numeric value used and compared with the upper and lower bounds of cost for such conditions [21].

Ideal Maintenance Cost (IMC) serves as the lower bound and the ideal cost in such maintenance applications. It is the incurred cost when the replacement action is performed one cycle before the failure, as shown in Equation (2). In this equation N denotes the number of equipment used for evaluation, $\mathbb{E}(T)$ is the expected failure state of the equipment.

$$\phi_{IMC} \approx \frac{N \cdot c_r}{N \cdot (\mathbb{E}(T) - 1)} \approx \frac{N \cdot c_r}{\sum_{j=1}^{N} (T_j - 1)} \tag{2}$$

Corrective Maintenance Cost (CMC) serves as the upper bound and the maximum cost in such maintenance applications. It is the incurred cost when the replacement action is performed after the equipment has failed as shown in Equation (3).

$$\phi_{CMC} \approx \frac{(c_r + c_f)}{\mathbb{E}(T)} \approx \frac{N \cdot (c_r + c_f)}{\sum_{j=1}^{N} T_j} \tag{3}$$

Average Optimal Cost $(\widetilde{Q^*})$ is the average cost that the agent receives as its performance on the test set as shown in Equation (4). In this equation $r(s,a)$ denotes the immediate reward as formulated in Equation (1), $Q^*(s',a')$ denotes the optimal action value of the next state-action pair, and γ is the discount factor.

$$\widetilde{Q^*}(s,a) = \frac{1}{N} \sum \left[r(s,a) + \gamma \max_{a'} Q^*(s',a') \right] \tag{4}$$

Average Remaining Useful Life $(\widetilde{RUL})$ before replacement It quantifies; how many useful cycles are remaining on average when the agent proposes the replacement action. Ideally, it should be one according to our defined criteria.

4   Proposed Methodology (SRLA)

The proposed hierarchical methodology integrates an IOHMM and a DRL agent. Within this hierarchical model, the purpose of the IOHMM is to identify when the system is approaching a desired (in our case: failure) state. Once the IOHMM has reached this failure state, the DRL agent's task is to optimize the decision on when to replace the equipment to maximize its total useful life. This IOHMM-DRL model allows for state- or event-based optimization. This further allows for a more efficient DRL training, since the training data set is restricted to the imminent-to-failure states. Figure 2 illustrates the proposed model which we name Specialized Reinforcement Learning Agent (SRLA).
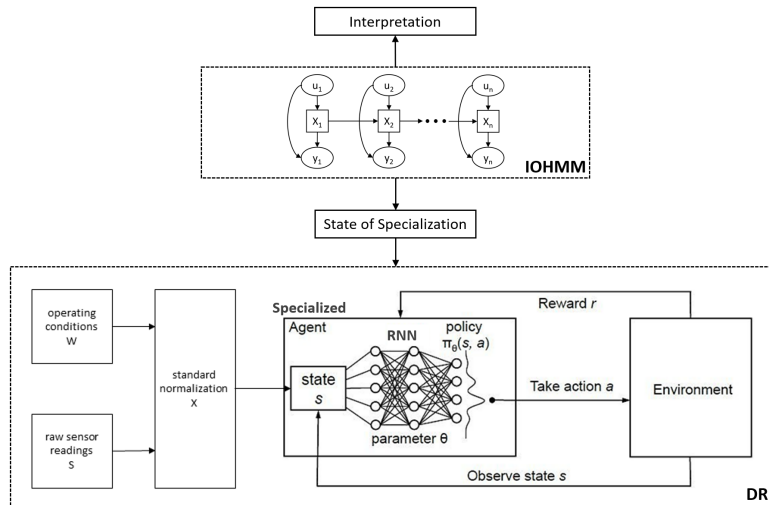
Fig. 2. Specialized Reinforcement Learning Agent (SRLA).

The DRL training and optimization process is relatively standard. We use Deep Learning (DL) as a function approximator that generalizes effectively to enormous state-action spaces through the approximation of unvisited states [3] as shown in Equation (5). In this equation L_i denotes the loss function, y_i is the TD target; which is the sum of the observed one-step reward and the discounted next Q (action) value conditioned on the current state and action, $Q(s,a)$ is the estimation of the Q value of the current state-action pair parameterized by θ.

$$
\begin{aligned}
L_i\left(\theta_i\right) &= \mathbb{E}_{a\sim\mu}\left[\left(y_i - Q\left(s,a;\theta_i\right)\right)^2\right]; \\
y_i &:= \mathbb{E}_{a'\sim\pi}\left[r + \gamma \max_{a'} Q\left(s',a';\theta_{i-1}\right) \mid S_t = s, A_t = a\right]
\end{aligned}
\tag{5}
$$

At a high level, an IOHMM is used, where the objective of training optimization is to identify the model parameters that best determine the given sequence of

observations conditioned on the given input. In the context of industrial settings, these inputs are the operating conditions that heavily influence the state of the system and control the system's behavior. Parameter γ is the vector defining the probability of being in each hidden state at a particular time $x_t = S_i$; given the input U, the observation sequence Y, and the parameters of the trained model λ (initial state, transition, and emission probability matrices conditioned on the input (U) as well), as shown in Equation (6). Parameter δ from Equation (7) in this context is used to predict the health degradation state sequence of the equipment, where the last cycle of each equipment determines the failure state. The inference algorithm for the SRLA is described in Algorithm A.1 of Appendix A.

$$\gamma_t(i) = P\left(x_t = S_i \mid U, Y, \lambda\right) \tag{6}$$

$$\delta_t(i) = \max_{x_1, \cdots, x_{t-1}} P\left[x_1 \cdots x_t = i, Y_1 \cdots Y_t \mid U, \lambda\right] \tag{7}$$

4.1   Interpretability with IOHMM

Beyond the performance considerations of the model, the IOHMM component provides a level of interpretability in terms of identifying failure states, the root cause of failure, and stages of health degradation. Based on the state sequence distributions predicted by the IOHMM from Equation (7), each state of a particular event can be decoded, such as the failure mode or degradation stage, as shown in [8]. To discover the most relevant sensor readings corresponding to these failure states that triggered the IOHMM to predict such a state, feature importance is performed that leads to the root cause failure analysis. Raw sensor readings are used as the input feature for the model and IOHMM state predictions are used as the target. After fitting the model, the importance of each sensor can be extracted for each IOHMM state. Apart from the failure event hypothesis, it is necessary to measure the health state of the equipment at different points to generate an alarm for the user when the equipment reaches a critical point of its lifetime. The interpretations are based on the critical points along the equipment degradation curve as shown in Figure 3 and the range of observed IOHMM states.
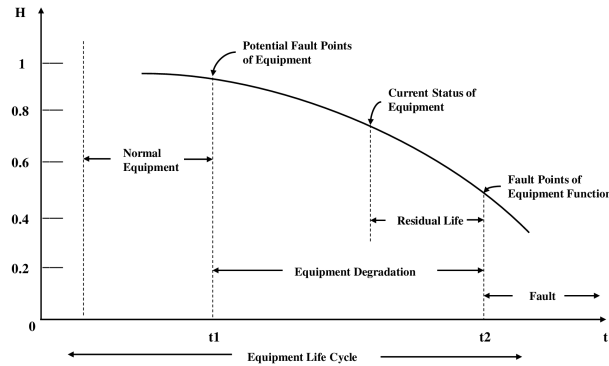

Fig. 3. Health degradation curve of equipment, taken from [12]

5   Experimental Setup

The two baseline systems defined in this paper are distinguished and designed by varying each of these four stages: (i) input, (ii) feature engineering, (iii) RL architecture, and (iv) output. The summary of the training parameters is shown in Appendix A.1 of Appendix A.

Baseline 1: Sensor Data + Operating Conditions: (i) Raw sensor data and operating conditions as the input, (ii) Standard normalization as the feature extraction module, (iii) DNN as the RL architecture, and (iv) Action policy at the output. It is used to set the failure cost to be used for the rest of the experiments.

Baseline 2: Sensor Data + Operating Conditions + IOHMM: (i) Raw sensor data and operating conditions as input, (ii) MinMax normalization and IOHMM as the feature engineering module, (iii) RNN as RL architecture, and (iv) Action policy, RUL estimation, and unsupervised clustering and interpretation based on events at output; as shown in Figure 4. Its significance is to determine the optimal number of IOHMM states to be used in the experiments. Implementation of IOHMM is done through a library [23]. This baseline uses the output of the IOHMM (probability distribution) as the input to the DRL agent, whereas SRLA uses the raw data as the input to the DRL agent during the state of specialization.
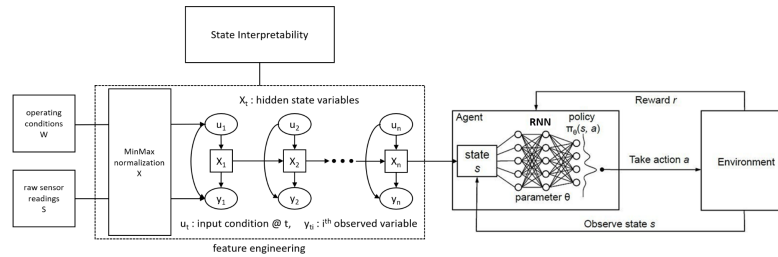
Fig. 4. IOHMM posterior probabilities as the input to DRL.

5.1   Setting the Hyperparameters for the Models

This section describes the experiments used to determine the hyperparameters (i) cost of failure (c_f) and (ii) number of IOHMM states. The effectiveness of the architectures has been evaluated as described in Section 3.3. The data set used for this part of the experiment is FD001, which is split into an 80:20 (train:test) ratio.

Calculating the cost of failure The reward function (Equation (1)) for the RL agent requires the specification of *cost of failure* (c_f) and *cost of replacement* (c_r). However, the NASA C-MAPSS data set does not specify these parameters.

To fix these values, we train Baseline 1 using a range of different c_f, while fixing c_r and then comparing and identifying the c_f that minimizes the average of total optimal cost per episode ($\widetilde{Q^*}$). c_r is fixed (100) and the comparison is based on the different c_f values of 25, 500 and 1000, as shown in Table 1. It was observed that as c_f increases, $\widetilde{Q^*}$ becomes closer to the ideal cost, and, at the same time, the number of failed units decreases to 0%. However, the agent becomes more cautious, suggesting replacement action earlier in the lifetime of the engine; thereby, increasing the average remaining cycles. In the context of predictive maintenance of safety-critical systems, it is more important to avoid failure at the expense of replacing equipment a few cycles before its remaining useful life. Therefore, c_f of 1000 was chosen for the rest of the experiments.

Calculating the number of hidden states Baseline 2 was used to find the number of states of the IOHMM model that maximizes the likelihood of our state space and the performance of the DRL through an iterative process. We evaluated the performance of the model as the number of states varied between 10, 15, and 20 states. The model trained through IOHMM gives the posterior probability distribution for every state as shown in Equation (6), which is then fed as an input to the DRL agent to be able to learn the optimal maintenance (replacement) policy. The experiment was carried out on the test set using the failure cost of 1000 and with the same parameters as the previous experiment for a better evaluation. 15 states of the IOHMM showed better performance results than the rest, and so in the rest of our experiment, we use 15 as the number of states for IOHMM model.

Table 1. Comparative evaluation and hyperparameter search.

Failure cost	Avg Q*	IMC	CMC	$\widetilde{RUL}$	Failed units
Baseline 1					
25	0.54	0.45	0.56	2.4	45%
500	0.61	0.45	2.68	7.5	5%
1000	0.49	0.45	4.92	7.0	0%
Baseline 2					
IOHMM states					
10	0.54	0.45	4.92	24.2	0%
15	0.49	0.45	4.92	6.8	0%
20	0.53	0.45	4.92	20.2	0%

6   Experiment 1: Interpretations Based on Hidden States

Data sets FD001, FD003, and DS01 are used in this section using the IOHMM for event-based hypothesis and state interpretations. The experiments performed here are to address the question of whether the introduction of the hidden states can help towards interpretability.

6.1   Interpretability - Failure Event Hypothesis

Due to the unavailability of the ground truth for other state mappings in FD003, just the failure states (last cycle state) were mapped in this experiment. Each failure state in the dataset is annotated with one of the 2 failure modes (HPC and fan degradation); however, the ground truth for the engines corresponding to which failure mode is not provided. Analyzing the failure states revealed two IOHMM states that corresponded to the failure event, which might be based on the two failure modes. To validate this hypothesis, the analysis was repeated with FD001, where there is only one failure mode defined in the description of the data set, and this analysis showed that only one IOHMM state was observed to be the failure state for each engine. This suggests that it is possible to map IOHMM states to failure events within the health state of the equipment.

Using the feature importance methodology described in Section 4.1, features (sensor readings) with a relatively higher score (based on feature importance) were selected from each class (failure states depicted by IOHMM). Further, the corresponding actual sensor information and description were extracted from [18] as described in Table 2. From the background information from the sensor descriptions, it was observed that the sensor importance for the two different IOHMM states showed a concrete failure event interpretation that corresponded to the failure described in the data set (HPC and Fan degradation), as hypothesized in Table 3.

Table 2. Feature to sensor description.

Feature	Sensor	Description
5	P_{30}	HPC outlet pressure
8	epr	Engine pressure ratio
10	phi	fuel flow : HPC pressure
13	BPR	Bypass ratio

Table 3. Sensor importance to failure event.

IOHMM state	Important sensor reading	Failure event hypothesis (interpretation)
9	BPR	Fan degradation
14	P_{30}, epr, phi	HPC degradation

6.2   Interpretability - State Decoding and Mapping

The second version of the NASA C-MAPSS data set [4] was used here to evaluate the state interpretability of IOHMM throughout the engine life, a subset of which is shown in Figure 5, where the red trend represents the IOHMM state prediction based on Equation (7). The data set has the ground truth values of the engine's state per cycle, the Boolean health state value is represented by the blue line with state 1 being healthy and 0 being unhealthy, the RUL is represented by the yellow line, and the green curve represents the health degradation curve. Based on the reference health degradation curve from Figure 3, and the range of IOHMM states observed during those conditions, we were able to associate different IOHMM states with different equipement conditions as shown in Table 4.
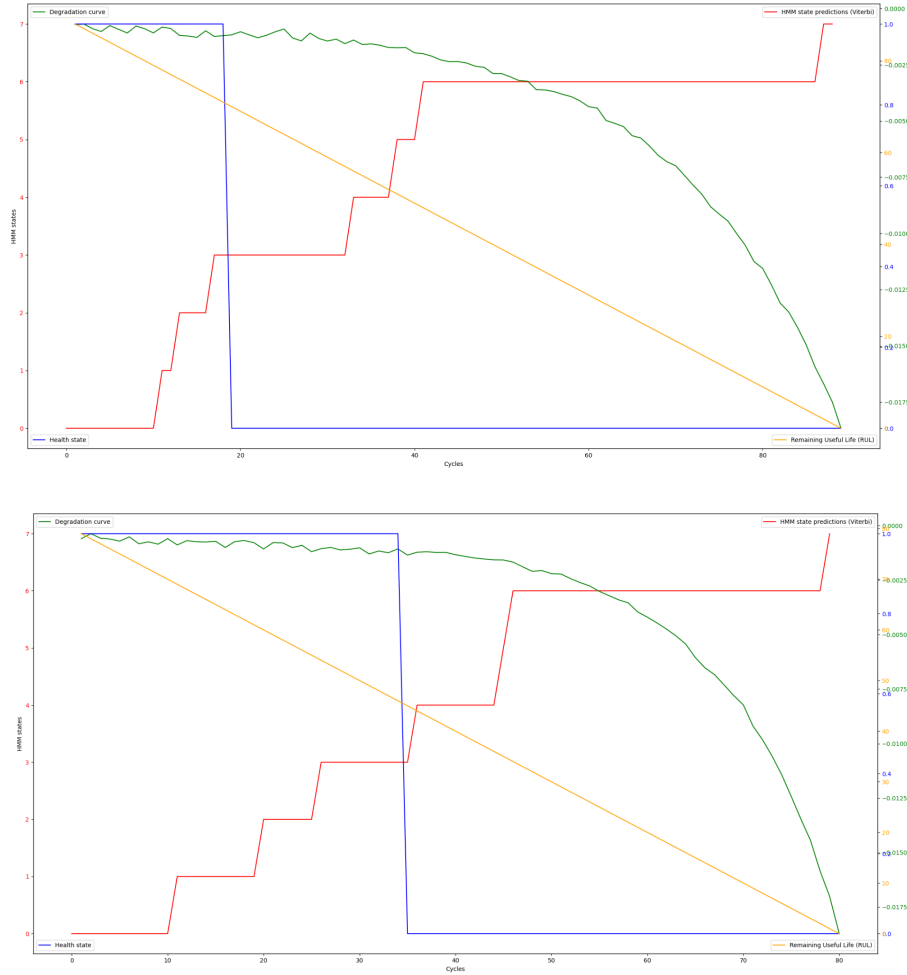
Fig. 5. State decoding and mapping for data set DS001.

Table 4. Interpretability of the IOHMM state to equipment conditions.

Equipment condition	IOHMM states
Normal equipment	0 - 2
Potential fault point of equipment	2 - 4
Failure progression	4 - 6
Fault point of equipment function	6 - 7
Failure	7

7 Experiment 2: Comparison of SRLA with Prior Work

Data set FD002 is used in this experiment for the comparative evaluation with baselines and prior work [21].

7.1 Comparative Evaluation and Results

As seen in Section 6.2, the IOHMM can align its states and state transitions with the relevant health states of the engine; however, the definition and alignment of the states were not fine enough to replace the engine with just one cycle before the failure. Therefore, DRL is used to refine the granularity after state distribution based on IOHMM, resulting in a hierarchical model. To evaluate the performance, the results are compared with the two baseline systems and the Particle Filtering (PF) based-DRL (*Bayesian particle filtering*) framework proposed by [5]. In their experiments [5] used 80 engines as the training set and 20 as the test set out of 260 engines. However, the engines were selected randomly; therefore, an exact comparison with the average agent cost could not be made. Therefore, the ratio of the Ideal Maintenance Cost (IMC) to the average agent cost ($\widetilde{Q^*}$) was compared in Table 5. As shown, Baseline 2 performs better than Baseline 1 and SRLA outperforms baseline systems and has a comparative performance with the PF + DRL methodology with the added benefits of interpretability.

Table 5. Comparison of the proposed methodology with baseline systems and [21].

Methodology	$\widetilde{Q^*}$	IMC	CMC	IMC/$\widetilde{Q^*}$	Failure	$\widetilde{RUL}$	Interpretations
Baseline 1	6.87	0.64	7.02	0.09	90%	2.6	No
Baseline 2	0.77	0.64	7.02	0.83	0%	23.0	Yes
PF + DRL [15]	2.02	1.93	20.80	0.96	0%	-	No
SRLA	0.69	0.64	7.02	0.94	0%	6.4	Yes

8   Conclusion and Future Direction

In this paper, a new hierarchical methodology was proposed utilizing the hidden Markov model-based deep reinforcement learning allowing the functionality of interpretability in the stochastic environment along with defining an optimal replacement policy and estimating remaining useful life without supervised annotations. Therefore, such a model can easily be used in industrial cases where the annotation of the fault type is difficult to obtain and the human supervisor in the loop can help define the state distribution according to the event-based analysis. To test the effectiveness of the model, the NASA C-MAPSS (turbofan engines) data sets versions 1 and 2 were used. It was compared with baseline models and prior work of Bayesian filtering-based-deep reinforcement learning to evaluate the performance. Our results indicate that the IOHMM-DRL framework outperforms the baseline DRL systems and has performance comparable to the Bayesian filtering DRL approach, with the added benefits of interpretability and a less complex system model. In the future, the proposed hierarchical architecture of IOHMM-DRL will be applied to other open data sets along with real-world case studies to measure its robustness.

References

1. Bengio, Y., Frasconi, P.: Input-output hmms for sequence processing. IEEE Transactions on Neural Networks **7**(5), 1231–1249 (1996). `https://doi.org/10.1109/72.536317`
2. Bengio, Y., Frasconi, P.: An input output hmm architecture. Advances in neural information processing systems pp. 427–434 (1995)
3. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-dynamic programming. Athena Scientific (1996)
4. Chao, A.: Manuel. et al. "Aircraft Engine Run-to-Failure Dataset under Real Flight Conditions for Prognostics and Diagnostics **6**, 1 (2021)
5. Chen, Z., et al.: Bayesian filtering: From kalman filters to particle filters, and beyond. Statistics **182**(1), 1–69 (2003)
6. Do, P., et al.: A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. Reliability Engineering & System Safety **133**, 22–32 (2015)
7. Dulac-Arnold, G., Levine, N., Mankowitz, D.J., Li, J., Paduraru, C., Gowal, S., Hester, T.: Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Machine Learning pp. 1–50 (2021)
8. Giantomassi, A., et al.: Hidden Markov model for health estimation and prognosis of turbofan engines. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference **5480** (2011)
9. Hofmann, P., Tashman, Z.: Hidden markov models and their application for predicting failure events. In: International Conference on Computational Science. pp. 464–477. Springer (2020)
10. Klingelschmidt, T., Weber, P., Simon, C., Theilliol, D., Peysson, F.: Fault diagnosis and prognosis by using input-output hidden markov models applied to a diesel generator. In: 2017 25th Mediterranean Conference on Control and Automation (MED). pp. 1326–1331 (2017). `https://doi.org/10.1109/MED.2017.7984302`
11. Lepenioti, K., et al.: Machine learning for predictive and prescriptive analytics of operational data in smart manufacturing. In: International Conference on Advanced Information Systems Engineering. , Cham (2020)
12. Li, H.Y., Xu, W., Cui, Y., Wang, Z., Xiao, M., Sun, Z.X.: Preventive maintenance decision model of urban transportation system equipment based on multi-control units. IEEE Access **8**, 15851–15869 (2019)
13. Meng, F., An, A., Li, E., Yang, S.: Adaptive event-based reinforcement learning control. In: 2019 Chinese Control And Decision Conference (CCDC). pp. 3471–3476. IEEE (2019)
14. Ong, K.S.H., Niyato, D., Yuen, C.: Predictive maintenance for edge-based sensor networks: A deep reinforcement learning approach. In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). pp. 1–6. IEEE (2020)
15. Panzer, M., Bender, B.: Deep reinforcement learning in production systems: a systematic literature review. International Journal of Production Research pp. 1–26 (2021)
16. Parra-Ullauri, J.M., García-Domínguez, A., Bencomo, N., Zheng, C., Zhen, C., Boubeta-Puig, J., Ortiz, G., Yang, S.: Event-driven temporal models for explanations-etemox: explaining reinforcement learning. Software and Systems Modeling pp. 1–23 (2021)
17. Rabiner, L., Juang, B.: An introduction to hidden markov models. IEEE ASSP Magazine **3**(1), 4–16 (1986). `https://doi.org/10.1109/MASSP.1986.1165342`

18. Saxena, A., Goebel, K.: Turbofan engine degradation simulation data set. NASA Ames Prognostics Data Repository : pp. 878–887 (2008)
19. Shahin, K.I., Simon, C., Weber, P.: Estimating iohmm parameters to compute remaining useful life of system. In: Proceedings of the 29th European Safety and Reliability Conference, Hannover, Germany. pp. 22–26 (2019)
20. Sikorska, J., Hodkiewicz, M., Ma, L.: Prognostic modelling options for remaining useful life estimation by industry. Mechanical systems and signal processing **25**(5), 1803–1836 (2011)
21. Skordilis, E., Moghaddass, R.: A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics. Computers & Industrial Engineering **147** (2020)
22. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
23. Yin, M., Silva, T.: Iohmm. `https://github.com/Mogeng/IOHMM` (2017)
24. Yoon, H.J., Lee, D., Hovakimyan, N.: Hidden markov model estimation-based q-learning for partially observable markov decision process. 2019 American Control Conference (ACC) (2019). `https://doi.org/10.23919/acc.2019.8814849`
25. Yoon, H.J., Lee, D., Hovakimyan, N.: Hidden markov model estimation-based q-learning for partially observable markov decision process. In: 2019 American Control Conference (ACC). pp. 2366–2371. IEEE (2019)

A Algorithms and Training Parameters

Algorithm A.1 Specialized Reinforcement Learning Agent (SRLA)

STEP I: IOHMM Training
Input:
n: number of hidden states
Y: output sequences
U: input seauences
Output: λ: model parameters (initial, transition, and emission probability)

STEP II: Viterbi Algorithm (IOHMM Inference)
Input: λ, U, Y
Output: $\delta_t(i) = \max_{x_1,\cdots,x_{t-1}} P\left[x_1 \cdots x_t = i, u_1 \cdots u_t, y_1 \cdots y_t \mid \lambda\right]$

STEP III: DRL Training
Input:
δ_s: specific event (such as failure)
S_t: $u_t + y_t$
Enviroment Modeling
Deep Reinforcement Learning
Output: $\hat{Q}^*(S_t, A_t)$

STEP IV: SRLA Inference
Input: λ, $\hat{Q}^*(S_t, A_t)$, S_t: (U_t, Y_t)
Step II, Interpretations Based on Hidden States
$\delta \rightarrow$ Specialized state $(X_s) \rightarrow U_s, Y_s$
if S_t in X_s **then**
    $\hat{Q}^*(s_t, a_t)$
    Environment Model
else
    $a_t =$ do nothing (hold)
end if
Output: $\hat{Q}^*(\delta_t, s_t, a_t)$

A.1 Training Parameters

The summary of the DL framework within the RL architectures is as follows: (a) Deep Neural Network (DNN) consisting of a total of 37,000 training parameters and fully-connected (dense) layers with 2 hidden layers that have 128 and 256 neurons, respectively, with ReLU activation. (b) Recurrent Neural Network (RNN) consists of 468,000 training parameters and fully connected (LSTM) layers with 2 hidden layers having 128 and 256 neurons, respectively. The output layer consists of the number of actions the agent can decide for decision-making with linear activation. The parameters of the DRL agent are as follows: discount rate = 0.95, learning rate = 1e-4, and the epsilon decay rate = 0.99 is selected with the initial epsilon = 0.5.