Equipment Health Indicator Learning Using Deep Reinforcement Learning

Chi Zhang(✉), Chetan Gupta, Ahmed Farahat, Kosta Ristovski, and Dipanjan Ghosh

Industrial AI Laboratory, Hitachi America Ltd., Santa Clara, CA, USA
{chi.zhang,chetan.gupta,ahmed.farahat,kosta.ristovski,
dipanjan.ghosh}@hal.hitachi.com

Abstract. Predictive Maintenance (PdM) is gaining popularity in industrial operations as it leverages the power of Machine Learning and Internet of Things (IoT) to predict the future health status of equipment. Health Indicator Learning (HIL) plays an important role in PdM as it learns a health curve representing the health conditions of equipment over time, so that health degradation is visually monitored and optimal planning can be performed accordingly to minimize the equipment downtime. However, HIL is a hard problem due to the fact that there is usually no way to access the actual health of the equipment during most of its operation. Traditionally, HIL is addressed by hand-crafting domain-specific performance indicators or through physical modeling, which is expensive and inapplicable for some industries. In this paper, we propose a purely data-driven approach for solving the HIL problem based on Deep Reinforcement Learning (DRL). Our key insight is that the HIL problem can be mapped to a credit assignment problem. Then DRL learns from failures by naturally backpropagating the credit of failures into intermediate states. In particular, given the observed time series of sensor, operating and event (failure) data, we learn a sequence of health indicators that represent the underlying health conditions of physical equipment. We demonstrate that the proposed methods significantly outperform the state-of-the-art methods for HIL and provide explainable insights about the equipment health. In addition, we propose the use of the learned health indicators to predict when the equipment is going to reach its end-of-life, and demonstrate how an explainable health curve is way more useful for a decision maker than a single-number prediction by a black-box model. The proposed approach has a great potential in a broader range of systems (e.g., economical and biological) as a general framework for the automatic learning of the underlying performance of complex systems.

Keywords: Health indicator learning
Deep Reinforcement Learning · Predictive Maintenance

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-10997-4_30) contains supplementary material, which is available to authorized users.

U. Brefeld et al. (Eds.): ECML PKDD 2018, LNAI 11053, pp. 488–504, 2019.
https://doi.org/10.1007/978-3-030-10997-4_30

1 Introduction

One of the important objectives in industrial operations is to minimize unexpected equipment failure. Unexpected downtime due to equipment failure can be very expensive – for example, an unexpected failure of a mining truck in the field can have a cost of up to $5000 per hour on operations. Besides the cost, unexpected failures can cause safety and environmental hazards and lead to loss of life and property. Traditionally industries have tried to address this problem through time-based maintenance. However, time-based maintenance often causes *over maintenance* [1], while still not fully being able to address the problem of unplanned downtime. With the advent of IoT, *Predictive Maintenance (PdM)* is gaining popularity. One of the key applications of PdM implies predicting the future health status of equipment (e.g., failure prediction), and then taking proactive action based on the predictions.

In the Machine Learning (ML) community, Predictive Maintenance (PdM) is typically modeled either as a problem of *Failure Prediction (FP)* [2] problem or the problem of estimating the *Remaining Useful Life (RUL)* [3], whereas in the Prognostics and Health Management (PHM) community, the problem is modeled as that of *Health Indicator Learning (HIL)*. FP answers the question (e.g., yes, no or in a probability) about whether a failure will occur in the next k days and RUL estimates the number of days l remaining before the equipment fails. HIL is the problem of estimating the future "health" $H(t)$ of the equipment, as a function of time t.

One reason for the popularity of RUL and FP in the ML community is that they are amenable to the traditional machine learning methods - FP is often modeled as a classification problem and RUL is modeled as a regression problem. RUL and FP modeled this way though useful, present operationalization challenges since ML produces black-box models and explainability is extremely desirable in industrial applications. Most domain experts are used to working with degradation curves and expect machine learning methods to produce something similar. Moreover, FP and RUL often do not provide enough information for optimal planning. For example, even if an operations manager is told that the equipment will fail in the next k days, they do not know whether to take the equipment offline tomorrow or on the k^{th} day, since the prediction provides no visibility into the health of the equipment during the k day period (i.e., explanatory power is missing). From a practical standpoint, solving these two problems simultaneously and independently often leads to inconsistent results: FP predicts a failure will happen in k days while RUL predicts a residual life of $l > k$ days.

HIL addresses most of these concerns. Since the output of HIL is a health curve $H(t)$, one can estimate the health of the equipment at any time t and observe the degradation over time. Moreover, once the health curve is obtained, both RUL and FP can be solved using the health curve in a mutually consistent manner. However, from a ML perspective, HIL is a hard problem. The problem is essentially that of function learning, with no ground truth - i.e., there is no way to observe the actual health of the equipment during most of the operation.

We observe the health only when the equipment fails, and typically most modern industrial equipment are reliable and do not fail very often.

Researchers in the PHM community address the HIL problem by either handcrafting a Key Performance Indicator (KPI) based on domain knowledge or try to identify some function of sensors that captures the health of the equipment through physical modeling. There are several challenges with these methods. Industrial equipment are complex (e.g., complicated and nested systems), and it is difficult and time-consuming for experts to come up with such KPIs. Additionally, domain-specific KPIs are not applicable across industries, so that developing a general method becomes infeasible. Furthermore, equipment is usually operated under varying operating conditions leading to significantly different health conditions and failure modes, making it difficult for a single manually-crafted KPI to capture health.

In this paper, we propose a machine learning based method to solve for HIL. Our key insight is that HIL can be modeled as a credit assignment problem which can be solved using Deep Reinforcement Learning (DRL). The life of equipment can be thought as a series of state transitions from a state that is "healthy" at the beginning to a state that is "completely unhealthy" when it fails. RL learns from failures by naturally backpropagating the credit of failures into intermediate states. In particular, given the observed time series of sensor, operating and event (failure) data, we learn a sequence of health indicators that represent the health conditions of equipment. The learned health indicators are then used to solve the RUL problem.

The contributions of this paper are summarized as follows:

- We propose to formalize the health indicator learning problem as a credit assignment problem and solve it with DRL-based approaches. To our knowledge, this is the first work formalizing and solving this problems within an RL framework. Additionally, the label sparsity problem (i.e., too few failures) is addressed due to the nature of RL.
- We propose a simple yet effective approach to automatically learn hyper parameters that are best for approximating health degradation behaviors. Therefore, the proposed method does not require domain-specific KPIs and are generic enough to be applied to equipment across industries.
- We use health indicators as compact representations (i.e., features) to solve the RUL problem, which is one of the most challenging problem in PdM. Therefore, we not only provide the explanation of health conditions for observed data, but can also predict the future health status of equipment.

The rest of the paper is organized as follows. Section 2 describes the details of the proposed method. Section 3 qualitatively analyzes the performance of the proposed methods on a synthetic problem. Section 4 shows experimental results applying our methods to a benchmark. Section 5 discusses the differences and relations between our method and other Markov chain based methods. Section 6 gives an overview of related work on HIL. Section 7 concludes the paper.

Table 1. Summary of notation

Notation	Meaning
$t \in \mathbb{N}$	Time step $t = 1, 2, ..., T$, where T is the last time step
$i \in \mathbb{N}$	Sequence index $i = 1, 2, ..., I$, where I is the total number of sequences
$x_i^t \in \mathbb{R}$	Sensor data x at time t from sequence i
$y_i^t \in \mathbb{R}$	Operating condition data y at time t from sequence i
$s \in \mathcal{S}$	State derived by discretizing $\{x_i^t\}$ into N bins: $s = Discretizing(x)$, where $Discretizing(\cdot)$ is a discretizing operation. $\mathcal{S} = \{s_n\}$ $n = 1, 2, ..., N$ is a finite set (i.e., state space)
$a \in \mathcal{A}$	Action defined as the change of operating conditions in two time steps: $a^t = \overrightarrow{\boldsymbol{u}^t} = c_y^{t+1} - c_y^t$, where $c_y = Clustering(y^t)$ is a clustering operation, $\boldsymbol{u}^t$ is a vector. $\mathcal{A} = \{a_m\}, m = 1, 2, ..., M$ is a finite set (i.e., action space)
$R^t \in \mathbb{R}$	Immediate reward at time t
$\pi(a\|s)$	Policy function of the probability of choosing action a given state s
$v^\pi(s)$	Value function of policy π
$U(s) \in \{0, 1\}$	Failure/non-failure labels where 1 denotes failure occurrence. (Failures indicate the end-of-life of equipment)
$\mathcal{P}_{s,a}(s')$	State transition probability function: when take action a in state s, the probability of transiting to state s'
$\mathcal{R}_{s,a}$	Reward function of the expected reward received after taking action a in state s
$\mathcal{M}$	MDP model $<\mathcal{S}, \mathcal{A}, \mathcal{P}_{\cdot,\cdot}(\cdot), \mathcal{R}_{\cdot,\cdot}, \gamma>$, where $\gamma \in [0, 1]$ is a discount factor

2 Methodology

In this section, we first formalize the health indicator learning as a credit assignment problem and propose to address it with RL in two ways: in the model-based method, a Markov Decision Process (MDP) is learned and used to derive the value function in a tabular form; in the model-free method, a bootstrapping algorithm with function approximation is used to learn a continuous value function without implicitly modeling the MDP. Consequently, the learned value function maps observation data of equipment to health indicators. The notations used in the paper are presented in Table 1.

2.1 Problem Formulation

The problem of HIL is to discover a health function $H(t) = v(s|s = s^t)$ such that the following two properties [5] are satisfied:

– *Property 1: Once an initial fault occurs, the trend of the health indicators should be monotonic*: $H(t) \geq H(t + \Delta), t = 1, 2, ..., T - \Delta$

– *Property 2: Despite of the operating conditions and failure modes, the variance of health indicator values at failures* $\sigma^2(H(T_i)), i = 1, 2, ..., I$ *should be minimized.*

where Δ is a small positive integer (e.g., $\Delta = 1$ means strictly monotonic), $v(s)$ is a function mapping the equipment state s at any time t to a health indicator value, and σ^2 represents the variance.

By assuming the health degradation process is a MDP, we can transform the HIL problem to a credit assignment problem and represent $v(s)$ as the value function (a.k.a., Bellman Equation) $v^\pi(s)$ in RL [6]:

$$v^\pi(s) = E^\pi[(R^{t+1}|S^t = s) + \gamma v^\pi(S^{t+1}|S^t = s)] \tag{1}$$

where γ is a discount factor, R^t is the reward function.

2.2 Model-Based Health Indicator Learning

According to [6], we assume conditional independence between state transitions and rewards in the model-based health indicator learning approach:

$$P[s^{t+1}, R^{t+1}|s^t, a^t] = P[s^{t+1}|s^t, a^t]P[R^{t+1}|s^t, a^t] \tag{2}$$

Learning the MDP Model: According to the definition of $\mathcal{M} = <\mathcal{S}, \mathcal{A}, \mathcal{P}_{\cdot,\cdot}(\cdot), \mathcal{R}_{\cdot,\cdot}, \gamma>$, now MDP learning can be transformed to a supervised learning problem, in which learning $\mathcal{R}_{s,a} =: s, a \rightarrow R$ is a regression problem and learning $\mathcal{P}_{s,a}(s') =: s, a \rightarrow s'$ is a density estimation problem. We use a Table Lookup method to learn $\mathcal{P}, \mathcal{R}$ directly, as presented in Algorithm 1.

Algorithm 1. MDP Learning

1: **procedure** DISCRETIZATION(Process sensor and operating condition data into states and actions)
2: $c_x \leftarrow Discretizing(\{x_i^t\})$
3: $c_y \leftarrow Clustering(\{y_i^t\})$
4: $\mathcal{S} = \{c_x\} = \{s\}$
5: $\mathcal{A} = \{c_y^t - c_y^{t+1}\} = \{a\}$
6: **procedure** LEARN $\mathcal{M}$ BY COUNTING VISITS $N(s, a)$ TO EACH (s, a) PAIR
7: $\mathcal{P}_{s,a}(s') = \frac{1}{N(s,a)} \sum_{i=1}^{I} \sum_{t=1}^{T_i} 1(s^t, a^t, s^{t+1} = s, a, s')$
8: $\mathcal{R}_{s,a} = \frac{1}{N(s,a)} \sum_{i=1}^{I} \sum_{t=1}^{T_i} 1(s^t, a^t = s, a)R^t$

A policy is defined as the probability of taking an action for a given state $\pi(a|s)$. Given that the objective of the proposed method is policy evaluation, the policy can be given or learned from data:

$$\pi(a|s) = \frac{1}{N(s)} \sum_{i=1}^{I} \sum_{t=1}^{T_i} 1(s^t, a^t = s, a) \tag{3}$$

where $N(s)$ is the number of occurrence of s. Note that stationary policies are assumed.

To define the immediate reward R^t, three scenarios (i.e., non-failure to non-failure, non-failure to failure, and failure to failure) are considered:

$$R^t = \begin{cases} 0 & U(s^t) = U(s^{t+1}) = 0 \\ -1.0 & U(s^t) = 0, U(s^{t+1}) = 1 \\ R_{ff} & U(s^t) = 1 \end{cases} \tag{4}$$

R_{ff} is a hyper-parameter that can be tuned and $R_{ff} > -1.0$ to impose more penalty for failure-to-failure transitions. We also propose an alternative reward function to handle faults and never-fail equipment in Appendix[1].

Learning Health Indicator Directly (HID). We rewrite Eq. 1 as:

$$\begin{aligned} v^\pi(s) &= E^\pi[R^{t+1}|S^t = s] + \gamma E^\pi[v(S^{t+1})|S^t = s] \\ &= \mathcal{R}(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_s(s') v(s') \end{aligned} \tag{5}$$

where $\mathcal{R}(s)$ and $\mathcal{P}_s(s')$ can be calculated as:

$$\mathcal{R}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}_{s,a} \tag{6}$$

$$\mathcal{P}_s(s') = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}_{s,a}(s') \tag{7}$$

Equation 5 can be expressed using matrices as:

$$\boldsymbol{v} = \boldsymbol{\mathcal{R}} + \gamma \boldsymbol{\mathcal{P}} \boldsymbol{v} \tag{8}$$

and solved as:

$$\boldsymbol{v} = (\boldsymbol{\mathcal{I}} - \gamma \boldsymbol{\mathcal{P}})^{-1} \boldsymbol{\mathcal{R}} \tag{9}$$

where $\boldsymbol{\mathcal{I}}$ is the identity matrix. For simplicity, we omit the notation π by using $v(s)$ to represent $v^\pi(s)$ throughout the paper without losing the meaning that $v(s)$ is the value function of the policy π. To deal with a large state space, we also propose an iterative approach to learn health indicator using Dynamic Programming (HIDP), as presented in the aforementioned Appendix.

2.3 Model-Free Methods

In real-world scenarios the equipment can go through a very large number of states, which makes the model-based method less efficient and inapplicable. Thus, we propose a model free method to learn the health indicators of states for a given policy *without* explicitly modeling π and $\mathcal{P}_{s,a}(s')$.

[1] https://tinyurl.com/yaguzvuc.

Algorithm 2. Temporal Difference (0) with Function Approximation

1: Initialize value function v with random weight θ
2: Build a memory of experience $e_i^t = (x_i^t, R_i^{t+1}, x_i^{t+1})$ into a data set $D = \{e_i^t\}, i = 1, ..., I, t = 1, ..., T_i$ and randomize.
3: **do**
4: Sample a minibatch of m experiences $(x, R, x') \sim U(D)$
5: **for** $i = 1, ..., m$ **do**
6: $z_i = R_i^{t+1} + \gamma \hat{v}_\theta(x_i^{t+1})$
7: $\theta := \arg\min_\theta(z_i - \hat{v}_\theta(x_i^t))$
8: **while** True

Health Indicator Temporal-Difference Learning with Function Approximation (HITDFA). To further extend the health indicator learning into the continuous space, we define $\hat{v}_\theta(x)$ parameterized by θ as a function to approximate the continuous state space, where θ can be a deep neural network. Given the objective function in Eq. 1, we use real experience instead of the expectation to update the value function, as shown in Algorithm 2. In the HITDFA method, memory replay [7] is used to remove correlations in the observation sequence and smoothing over changes in the data distribution.

Note that the proposed HITDFA method learns the health indicators using value function $v(s)$, but it can be easily extended to learn the action value function $q(s, a)$ for each state-action pair. Consequently, we can replace $\hat{v}_\theta(x_i^t)$ with $\hat{q}_\theta(x_i^t, a_i^t)$ and $\hat{v}_\theta(x_i^{t+1})$ with $\hat{q}_\theta(x_i^{t+1}, a_i^{t+1})$. Similarly, such extension can also be applied to HID method.

In the proposed RL-based methods, there are two hyper-parameters that need to be learned: γ and R_{ff} (Eq. 4). We use a simple yet effective grid search approach to find the hyper parameter settings that (1) make $H(t)$ monotonic (i.e., the first constraint in Sect. 2.1), and (2) obtain the minimum variance of $H(T_i), i = 1, 2, ..., I$ (i.e., second constraint).

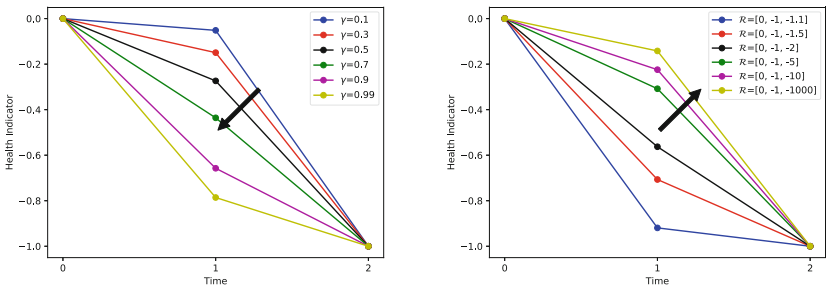
3 Qualitative Analysis of RL-Based HIL

In this section, we first study an ideal equipment which has only three states $\{s_1, s_2, s_3\}$, representing health, intermediate, and failure states respectively. The initial state $s^0 = s_1$ and the state probability transition matrix is given as:

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

Now we use HID to learn the health indicators. By selecting the reward function $R_{ff} = -2$ and varying γ, various $\boldsymbol{v}$ can be calculated according to Eq. 9. Since the objective is to study the degradation behaviors, all the learned value functions are rescaled to $[-1, 0]$ with 0 indicating healthy and -1 indicating failed, as presented in Fig. 1(a). When we fix $\gamma = 0.9$ and vary R_{ff}, significantly

different degradation behaviors are obtained. It can be observed in Fig. 1(b) that a larger γ leads to the concave property, which is because γ as the discount factor trades-off the importance of earlier versus later rewards. Therefore, a larger γ tends to emphasize the affect of failures happening at the end of the sequence. The opposite trend is observed when the immediate reward of staying in a failure state is penalized much more than transferring to a failure state (i.e., more negative R_{ff}). This is because the reward shapes the underlying relationship between failure and non-failure states. A more negative reward implies the larger difference between health indicator values at failure state and non-failure state, which makes the curve in Fig. 1(b) shift from concave to convex.



(a) Effect of γ (b) Effect of R_{ff}

Fig. 1. Using γ and R_{ff} to characterize different health degradation behaviors.

Therefore, the proposed method is capable of approximating different health degradation behaviors by tuning hyper parameters to represent different relationships between failure/non-failure states for a given problem (i.e., MDP). This also differentiates our methods from previous works such as [4] that can only model one type of degradation behavior (e.g., exponential). Potentially, the proposed methods are applicable to a wide range of physical equipment.

4 Quantitative Analysis of RL-Based HIL

To evaluate the performance of the proposed learning methods, a benchmark is used to demonstrate that the learned health indicators satisfy essential properties in health degradation development (i.e., Properties 1 and 2 in Sect. 2.1). Then, we conduct experiments to validate the effectiveness of the proposed method by examining if testing health indicator curves in run-to-failure data fall into the same failure threshold estimated from training curves. Lastly, our approaches are combined with a regression model to predict RULs to demonstrate that different PdM tasks can be achieved using the proposed methods.

4.1 Experimental Setup

To quantitatively investigate the effectiveness of the proposed methods, we conduct experiments with the NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) data set, as it is a standard benchmark widely used in PHM community [9]. This is a Turbofan Engine Degradation Simulation Data Set contains measurements that simulate the degradation of several turbofan engines under different operating conditions. This benchmark has four data sets (FD001 $\sim$ FD004), consisting of sensor data, operating condition data, and models a number of failure modes. At each operating time cycle, a snapshot of sensor data (i.e., 21 sensors) and operating condition data (i.e., 3 operating conditions) are collected. The sensor readings reflect the current status of the engine, while the operating conditions substantially effect the engine health performance. More information about this data set can be found in [9]. The engine is operating normally at the start of each time series, and develops a failure at some point in time, since which the fault grows in magnitude until failure. Before using the data set, we preprocess each sensor dimension data by MinMax normalization $x_i = \frac{x_i - \min(x_i)}{\max(x_i) - x_i}$. The proposed methods are implemented on a Linux machine with an Intel Xeon E5 1.7G 6-core CPU, an NVIDIA TITAN X GPU, and 32 Gb memory. For HID and HITDFA, both the health indicator inference time and prediction time are in a few milliseconds level, which are far less than one operating cycle in most human-operated equipment.

As described in [9], the operating conditions can be clustered into 6 distinct clusters. We apply K-means to cluster all operating condition data and construct the action set $\{a\} = \mathcal{A}$ as defined in Sect. 2. The sensor data is clustered into 400 distinct states. For each data set, we assume a unique policy and learn the corresponding value function.

Using the C-MAPSS data set, we quantitatively evaluate our methods in the following tasks:

- **Health Degradation Behavior Analysis:** Given run-to-failure data, validate the soundness of the proposed methods by examining $H(t)$ with the two essential properties and an additional property that is given in the data set, and compare the results with baselines.
- **HIL Performance Validation:** Given run-to-failure data, split the set into training and testing sets, then learn $H_{train}(t)$ on the training set to derive the distribution of the failure threshold, and finally validate on the testing set that $H_{test}(T_i)$ falls into the same threshold.
- **RUL Estimation:** Given run-to-failure data and prior-to-failure data, train a regression model based on $H_{train}(t)$ in the run-to-failure data, and use the learned regression model to predict $H_{test}(t)$ in the prior-to-failure set to estimate RULs.

4.2 Health Degradation Behavior Analysis

Using the proposed approach in Sect. 2.3, the hyper parameters satisfying the constraints for each policy (i.e., data set) are found. The health indicators of

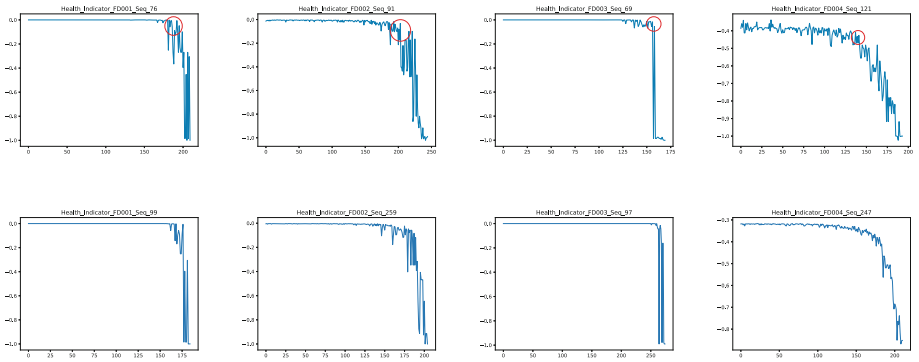


Fig. 2. Health degradation of randomly selected sequences learned by HID. First row: training, second row: testing. (Color figure online)

randomly selected training and testing sequences from each sub data set is presented in Fig. 2. It can be observed that even though there are fluctuations (due to nosies and can be smoothed by postprocessing), the monotonic property still holds (i.e., the first constraint in Sect. 2.1 is satisfied). For different $H(t)$ curves, the critical points (as highlighted by the red circles in the first row in Fig. 2) after which the value decreases sharply are also observed. A plausible explanation is that the proposed method learns the health value of each state, and the state transition in the sequence decides the shape of the curve, as well as the critical points in the curve.

To validate the second constraint in Sect. 2.1, we compare results of the learned model with a baseline health indicator learning method [4] in Table 2. Our proposed methods significantly outperform the composite health indicator and sensor-wise indicators in orders of magnitude. Note that [4] only studies FD001 set, so that we compare variances on the same data set.

Even thought the ground truth health values are not available, Saxena et al. [9] reveals that the degradation is exponential: $H(t) = 1 - d - exp\{at^b\}$, where d is the non-zero initial degradation and a, b are coefficients. It can be clearly observed from Fig. 2 that our model is able to characterize the exponential form of degradation.

Table 2. Comparison of variances between proposed methods and baselines. From T24 to W32, and CompHI are health indicators in [4], and the variance results are given by using the corresponding sensor.

Hea. Ind.	T24	T50	P30	Nf	Ps30	Phi	NRf
Variance	0.0274	0.014	0.0264	0.0683	0.0154	0.0206	0.0580
Hea. Ind.	BPR	htBleed	W31	W32	CompHI	HID	HITDFA
Variance	0.0225	0.0435	0.0220	0.0317	0.0101	7.1×10^{-5}	1.05×10^{-5}

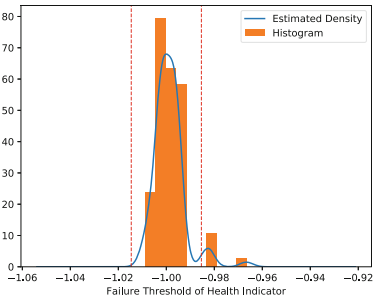


Fig. 3. Histogram and density estimation of $H(T_i)$ (FD001). (Color figure online)

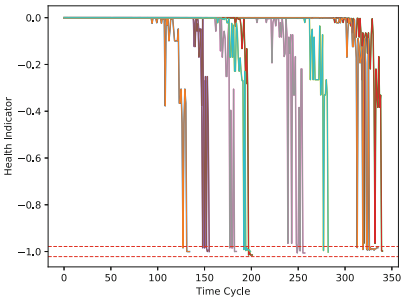


Fig. 4. Randomly selected $H(t)$ for different engines learned by HID (FD001). (Color figure online)

4.3 HIL Generalization Performance Validation

It is important to validate that when apply the learned model to unseen (i.e., testing) run-to-failure sequences, it is still capable of rendering consistent health degradation behavior and ends up with consistent health indicator values at the failures. In this section, we conduct such experiments to validate the generalization performance of the proposed methods. The run-to-failure data in each data set is split into training (90%) and testing (10%) sets. In training, we apply HID method to learn $v(s)$ for each state, and obtain $H_{train}(t)$. The corresponding health indicator values at failures (i.e., $\{h_T\}_{train} = H_{train}(T_i), i = 1, 2, ...I$) are obtained as well. Note that we assume in each data set, testing data and training data are generated under the same policy.

We estimate the density function $F(h_T)$ of h_T using Gaussian kernel (as shown in Fig. 3). Then the failure threshold h_f is defined as $h_f = argmax(F(h_T))$. As suggested by [9], health scores are normalized to a fixed scale, we also normalize health indicators by h_f: $H(t) = \frac{H(t)}{|h_f|}$, so that $h_f = -1$. $H(t)$ is rescaled to a region close to $[-1, 0]$, where 0 represents the perfect health status and -1 represents the failed status. In Fig. 3, the area between the red dash lines (with the right line indicates $h_{f_{max}}$ and left line indicates $h_{f_{min}}$) indicating the confidence level c (e.g., 95%): $c = \int_{h_{f_{min}}}^{h_{f_{max}}} F(h_T)dh_T$.

In testing, we apply the learned model to obtain $H_{test}(t)$, and $\{h_T\}_{test} = H_{test}(T_i)$. By finding the health indicator that is closest to h_f, we can obtain the estimated failure time: $T' = H^{-1}(\arg\max(F(h_T)))$, where $h_{f_{min}} \leq h_T \leq h_{f_{max}}$. Figure 4 presents $H(t)$ of randomly selected testing data and the learned failure threshold region, where the red dash lines correspond to the confidence area in Fig. 3. We use Root-Mean-Square-Error (RMSE) for performance evaluation: $RMSE = \sqrt{\frac{1}{I}\sum_{i \in I}(T_i' - T_i)^2}$.

It can be observed in Fig. 5 that our approach achieves better performance on FD001 and FD003 compared to FD002 and FD004, due to the fact that FD001 and FD003 have a simpler policy (i.e., $\pi(a_0|s_i) = 1, i = 1, 2, ..., N$, where a_0 is the action that is always taken), and FD002 and FD004 involve complicated policies

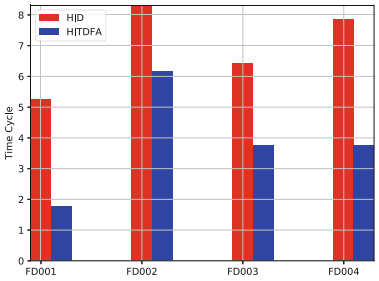


Fig. 5. Average error between actual and estimated time of failures in time cycles in the testing data.

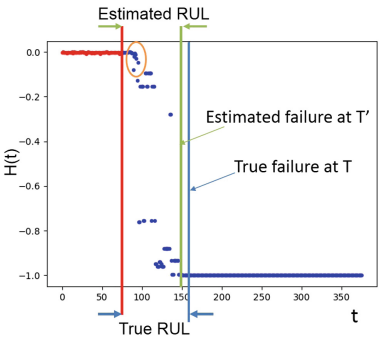


Fig. 6. RUL estimation. (Color figure online)

that are difficult to evaluate. It can also be observed that HITDFA has a lower variance than HID, due to the fact that (1) model-free method can yield better performance since its performance does not depend on the quality of a MDP model; (2) function approximation is adopted in HITDFA so that information lost in state discretization in the HID is avoided.

The objective of this experiment is to validate that the proposed methods is powerful in characterizing the health curves when new observations are given. Our proposed health indicator learning methods can be easily combined with regression models for prediction, as presented in the next section.

4.4 Remaining Useful Life Estimation

In this experiment, we use the health indicators learned from our proposed methods to solve the RUL problem. The objective is to validate that our methods are capable of learning a good representation of health conditions, which can be used as a feature for prediction models to solve PdM problems.

We use all run-to-failure data as a training set, and prior-to-failure data as a testing set. First, we learn the value function from the training set, and derive the health indicator sequences $H_{train}(t)$ and $H_{test}(t)$. Then $H_{train}(t)$ are used to learn a Seq2Seq deep neural network [8], which can predict health indicator sequence $\hat{H}(t)$ given $H_{test}(t)$ derived from prior-to-failure data. Finally, the failure threshold in Sect. 4.3 is applied to find the failure time T' in $\hat{H}(t)$. Figure 6 shows the given health indicators (red dots), the predicted health indicators (blue dots), and the true and predicted failures. The predicted health degradation curve provides rich information that can be easily interpreted by operators. For example, the critical points (in orange circle) can be used to decide when to perform preventive maintenance on the equipment to avoid downtime and maximize utilization.

In Table 3, we compare *RMSE* of our methods with state-of-the-art Deep Learning (DL) approaches. It can be observed that our methods achieve

performance worse than LSTM and CNN, and better than MLP. A plausible explanation is that the learned health indicator is actually a 1D compact representation of sensor measurement from equipment, which is semantically meaningful (i.e., health), but certain information is compressed or lost. In contrast, DL approaches map sensor measurements to RULs directly by constructing complex representations [3] so that the lower error rate are achieved. However, DL-based RUL estimation can only give a single number about how many days left before failure, without any insight about the health degradation over time. Therefore, it is difficult to make the optimal operation and maintenance decisions solely based on the RULs predicted by these approaches.

Table 3. RMSE (%) comparison of RUL with DL methods on C-MAPSS data

Data Set	#1	#2	#3	#4	Average
MLP (Multi Layer Perceptron) [12]	37.56	80.03	37.39	77.37	58.09
SVR (Support Vector Regression) [12]	20.96	42.00	21.05	45.35	32.34
RVR (Relevance Vector Regression) [12]	23.80	31.30	22.37	34.34	27.95
CNN (Convolutional Neural Network) [12]	18.45	30.29	19.82	29.16	24.43
Deep LSTM (Long-short Term Memory) [3]	16.14	24.49	16.18	28.17	21.25
LSTMBS [13]	14.89	26.86	15.11	27.11	20.99
HID	23.76	38.80	37.11	58.16	39.46
HITDFA	23.01	39.00	39.69	58.10	39.95

Besides of the comparison with DL, we are more interested in comparing our methods with other HIL approaches. In these approaches, Mean-Absolute-Percentage-Errors ($MAPE$) is the most popular metric as it measures the relative - rather than the absolute - magnitude of the prediction error in sequences. Therefore, we use the same $MAPE = \frac{1}{I}\sum_{i\in I}\left|\frac{T_i'-T_i}{T_i+O_i}\right|$ as [4], where O_i is the length of observed sequence i. From Table 4, it can be observed that our methods outperform state-of-the-art HIL approaches. The improvement is calculated as $Improvement = 1 - (MAPE_{HID}/MAPE_{state-of-the-art})$. It is noteworthy that our methods achieve a good performance even in FD002 and FD004, which are known to be difficult to learn due to the complex operating conditions and failure modes in the data. A plausible explanation is that we regard each data set independently to learn and evaluate the corresponding policy, so that the interferences in various policies are excluded. Ideally, if the environment used to generate the data (e.g., a simulator) is available, a random policy can be applied to acquire infinite experience to learn $v(s)$, despite of multiple or single operating conditions and failure modes.

Table 4. Comparison of the proposed methods with other HIL methods on C-MAPSS data based on MAPE (%) (table adapted from [18])

Data Set	#1	#2	#3	#4	Average
CompHI [4]	9	-	-	-	-
ANN [10]	43	48	44	49	46
FS-LSSVR [10]	38	44	40	47	42.5
RULCLIPPER [11]	20	32	23	34	27.25
HID	9.87	**18.05**	**13.08**	**31.00**	**18.00**
HITDFA	**8.85**	18.10	14.10	32.00	18.27
Improvement	1.67%	43.59%	43.13%	8.82%	33.95%

5 Discussion

In this section, we discuss relationship of proposed methods with Markov Chain and Hidden Markov Model. For equipment without explicitly defined controllable parts (i.e., no actions), the proposed MDP model can be easily transformed to Markov Chain by removing a^t and the proposed methods can still be applied. Our methods are also different from Hidden Markov Models (HMMs). First, HMM is supervised and requires the actual health indicator values for training, while our RL-based method does not. In HMM-based approaches, it usually creates X labels artificially by defining different health levels. The coarse definition of health levels leads to a very rough HIL results. Moreover, the definition and segmentation of health levels require a lot of domain knowledge. In contrast, the proposed method discovers the evolution of health indicators automatically by learning values for $N(>> X)$ discrete sensory states. Second, our method models actions separately from state, while HMM considers actions as some additional dimensions in the state space. This leads to $N \times M$ states (M is the number of actions) and X hidden states and hence, makes the computation expensive. In contrast, our method only has N states and no hidden states.

When generalize the proposed methods to other domains where the feature spaces are large (as opposed to 21 sensor data and 3 operating conditions in C-MAPSS), feature selection approaches can be used to preprocess the data in HIT. Automatic feature learning mechanism such as convolutional layers can be used as the first part in a deep network (i.e., θ) in HITDFA.

6 Related Work

In this section, we first give an overview of methods formalizing the health indicator learning problem as a classification problem and learning coarse-grained health indicators. Then we review methods learning fine-grained health values by mapping sensor measurement to continuous health values. Lastly, methods that map the learned health indicators to RULs to perform prognostics are reviewed.

To detect anomaly, assess health, or identify failures, a known target (i.e., ground truth) is required to evaluate the performance of the learned health indicators. Some methods [14,15] label data with pre-defined degradation levels: normal, knee corresponding, accelerated degradation, and failure. Then the learning can be converted to a classification problem that can be solved using supervised approaches such as HMM [14]. This type of methods require hand-crafted segmentations in order to label the data with the levels, which heavily depends on domain knowledge, and these labels are only approximation to ground truth, which all make it less applicable in general health indicator learning. Different from these methods, our approach is capable of automatically learning health indicators from data, without relying on hand-picked labels.

Due to that classification approaches can only learn coarse health levels, alternative approaches are proposed to map sensor measurements to continuous health indicator values. By leveraging dimension reduction methods, the work of [16] finds the most informative individual sensor and learns a health index. In contrast, in this paper we proposed to learn a mapping from all sensor measurement to health indicators, without losing any information in dimension reduction as in [16]. Similar to our methods, a composite health index is modeled by fusing data from all sensors [4]. However, the work of [4] can only deal with equipment operated under a single operating condition and falls into a single failure mode, while our proposed methods can handle more complicated situations.

Based on the learned health index, the prognostics problem is addressed by learning a second mapping links health index values to the RUL. Le Son et al. [17] models the health degradation process as a gamma process, then finds the hidden degradation states by Gibbs algorithm, and estimates RUL as a random variables with a probability distribution. As discussed in [18], the performance for RUL prediction depends on both the health indicator learning and prediction.

7 Conclusion

In the emerging area of predictive maintenance, there is a crucial demand from the users to abstract the complexity of physical systems behind explainable indicators that reflect the true status of these systems and justify the very costly actions that the users might take in response to predictions. This paper takes a major step forward toward achieving this goal by providing the ability to learn health indicators from sensor data given information about a few failure incidents. To the best of our knowledge, this is the first work to formulate this Health Indicator Learning (HIL) problem as a credit assignment problem and model the health indicator as the output of a state value function. The paper proposed both model-based and model-free RL methods to solve the value function. In particular, we proposed an automatic hyperparameter learning approach by using simple physical properties as constraints, which makes our method widely applicable across different domains and industries. We demonstrated the effectiveness of our method on synthetic data as well as well-known benchmark data

sets. We showed that the method can learn the health indicators of equipment operating under various conditions even in the presence of data from various failure modes. Experiments also demonstrated that the proposed methods achieve 33.95% improvement in predicting Remaining Useful Life (RUL) in comparison to state-of-the-art methods for the HIL problem. Our method keeps the distinct quality of HIL methods in providing explainable predictions in the format of a degradation curve. This is in contrast to black-box regression models such as LSTM which directly predict RUL as a single number and provide a complex feature map that cannot be explained to decision makers.

References

1. Ahmad, R., Kamaruddin, S.: An overview of time-based and condition-based maintenance in industrial application. CAIE **63**(1), 135–149 (2012)
2. Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A.: Machine learning for predictive maintenance: a multiple classifier approach. IINF **11**(3), 812–820 (2015)
3. Zheng, S., Ristovski, K., Farahat, A., Gupta, C.: Long short-term memory network for remaining useful life estimation. In: IEEE PHM (2017)
4. Liu, K., Gebraeel, N.Z., Shi, J.: A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. ASE **10**, 652–664 (2013)
5. Saxena, A., et al.: Metrics for evaluating performance of prognostic techniques. In: IEEE PHM (2008)
6. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
7. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
8. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)
9. Saxena, A., Goebel, K., Simon, D., Eklund, N.: Damage propagation modeling for aircraft engine run-to-failure simulation. In: IEEE PHM (2008)
10. Li, X., Qian, J., Wang, G.G.: Fault prognostic based on hybrid method of state judgment and regression. AIME **5**, 149562 (2013)
11. Ramasso, E.: Investigating computational geometry for failure prognostics in presence of imprecise health indicator: results and comparisons on C-MAPSS datasets. PHM **5**(4), 005 (2014)
12. Sateesh Babu, G., Zhao, P., Li, X.-L.: Deep convolutional neural network based regression approach for estimation of remaining useful life. In: Navathe, S.B., Wu, W., Shekhar, S., Du, X., Wang, X.S., Xiong, H. (eds.) DASFAA 2016. LNCS, vol. 9642, pp. 214–228. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32025-0_14
13. Liao, Y., Zhang, L., Liu, C.: Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method. In: IEEE PHM (2018)
14. Ramasso, E.: Contribution of belief functions to Hidden Markov models with an application to fault diagnosis. In: MLSP (2009)
15. Ramasso, E., Denoeux, T.: Making use of partial knowledge about hidden states in HMMs: an approach based on belief functions. FUZZ **22**(2), 395–405 (2014)

16. El-Koujok, M., Gouriveau, R., Zerhouni, N.: Reducing arbitrary choices in model building for prognostics: an approach by applying parsimony principle on an evolving neuro-fuzzy system. Microelectron. Reliab. **51**(2), 310–320 (2011)
17. Le Son, K., Fouladirad, M., Barros, A.: Remaining useful life estimation on the non-homogenous gamma with noise deterioration based on gibbs filtering: a case study. In: IEEE PHM (2012)
18. Ramasso, E., Saxena, A.: Performance benchmarking and analysis of prognostic methods for CMAPSS datasets. PHM **5**(2), 1–15 (2014)