Artificial Intelligence Review (2023) 56:12885–12947
https://doi.org/10.1007/s10462-023-10468-6

Reinforcement learning for predictive maintenance: a systematic technical review

Rajesh Siraskar[1,3] · Satish Kumar[1,2] · Shruti Patil[1,2] · Arunkumar Bongale[1] · Ketan Kotecha[1,2]

Accepted: 9 March 2023 / Published online: 25 March 2023

Abstract

The manufacturing world is subject to ever-increasing cost optimization pressures. Maintenance adds to cost *and* disrupts production; optimized maintenance is therefore of utmost interest. As an autonomous learning mechanism reinforcement learning (RL) is increasingly used to solve complex tasks. While designing an optimal, model-free RL solution for predictive maintenance (PdM) is an attractive proposition, there are several key steps and design elements to be considered—from modeling degradation of the physical equipment to creating RL formulations. In this article, we survey how researchers have applied RL to optimally predict maintenance in diverse forms—from early diagnosis to computing a "health index" to *directly* suggesting a maintenance action. Contributions of this article include developing a taxonomy for PdM techniques in general and one specifically for RL applied to PdM. We discovered and studied unique techniques and applications by applying $tf - idf$ (a text mining technique). Furthermore, we systematically studied how researchers have *mathematically* formulated RL concepts and included some detailed case-studies that help demonstrate the complete flow of applying RL to PdM. Finally, in Sect. 14, we summarize the insights for researchers, and for the industrial practitioner we lay out a simple approach for implementing RL for PdM.

Keywords Reinforcement learning · Predictive maintenance · Taxonomy · Practitioner · Case-studies · Mathematical treatment

✉ Satish Kumar
satishkumar.vc@gmail.com

Rajesh Siraskar
rajesh.siraskar@gmail.com

[1] Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India

[2] Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India

[3] Birlasoft Ltd., Pune 411057, Maharashtra, India

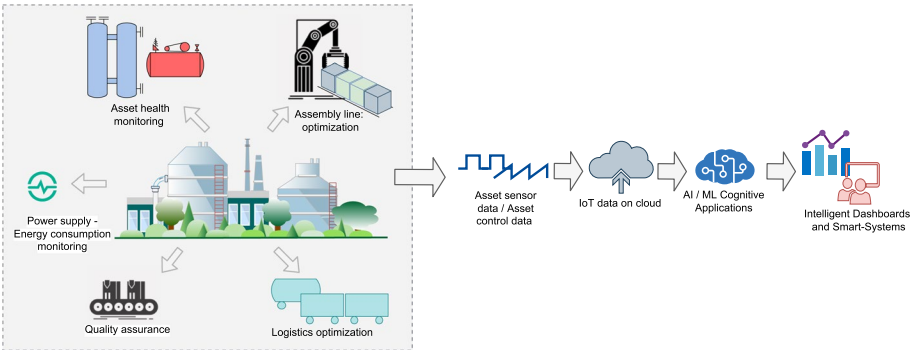


Fig. 1 SmartFactory operations

1 Introduction

Industry 4.0 is defined as the application of Internet of Things (IoT) to drive *intelligent action* back to the physical world (Deloitte 2020). IoT devices with embedded sensors continue to advance technologically and yet become cheaper. As of 2021, Ericsson (2021) reports 14.6 B connected devices world-wide while Dahlqvist et al. (2019) predicts 43 B by 2023.[1] Real-time analytics, with sensors at the "edge" (installed on the machines), combined with artificial intelligence (AI), are shifting activity to more powerful platforms on the cloud. With the advent of 5 G high-speed connectivity, really complex AI applications can begin to show impact and assist industrial operations and create effective cyber-physical systems.

Burke et al. (2017) defines SmartFactory as a system that *self-optimizes* performance—learning from new conditions and adapting in near-real time. Predictive maintenance (PdM) utilizes data from multiple sources, such as sensors installed on critical equipment, information from enterprise resource planning (ERP) systems, production data and maintenance information from maintenance management systems. SmartFactory management systems, Fig. 1, use data along with advanced prediction models to predict failures and *proactively* address them.

Ben-Daya et al. (2012) attributes 15% to, as high as 40%, of the manufacturing costs to maintenance; this was across several industries. Despite advantages there is reluctance as Laape et al. (2020) reports nearly 65% manufacturers surveyed did not report progress on SmartFactory initiatives. Manufacturing industry faces tremendous cost pressures and *optimal predictive maintenance* will allow industries to maintain equipment to ensure continuous production as well as lower maintenance spend.

Researchers have applied a wide variety of techniques to solve PdM problems—from mixed-integer and multi-criteria optimization techniques such as Pareto optimization (Saydam and Frangopol 2015), to machine learning (ML) by applying decision trees (event trees) (Frangopol et al. 1997), random forests (Kabir et al. 2018) and gradient boosted trees (XGboost) (Ma et al. 2020), to unsupervised ML techniques, like PCA (Eke et al. 2017; Susto et al. 2014), to supervised techniques like regression (Susto et al. 2014) and support

[1] Sinha (2021) reports 12.3 B devices as of 2021 and estimates 27.0 B to be connected by 2025.

vector machines (SVM) (Ding et al. 2008; Susto et al. 2013). Deep learning techniques like long short term memory (LSTM) have been applied by Sayyad et al. (2022); Zheng et al. (2017a) for RUL (remaining useful life) estimation while Sateesh Babu et al. (2016) applied regression using convolutional neural network (CNN) for the same. Finally, Skydt et al. (2021) perform early fault-detection as a trigger for PdM.

Pure statistical representations of systems and use of deterministic methods limited the effectiveness of early methods. Introduction of ML helped better modeling of systems by use of data from the systems, however having enough data labeled by experts, is always a challenge and leads to approximate solutions.

Reinforcement learning (RL) is an autonomous learning mechanism and in principle, overcomes some of the above limitations. Traditional methods of designing control and planning systems are challenged by complex nonlinear processes and RL is increasingly proving to be a more effective method to build model-free optimal solutions (Lewis et al. 2012; Sutton and Barto 2018). PdM is an optimal planning problem and this article surveys the research done in applying RL for solving it. Note that, in this article the term RL encompasses deep RL as well.

2 Motivation for our research

Application of RL to the field of PdM have not been surveyed exhaustively. Only 8 review articles are available.[2] Table 1 shows a comparison.

Barja-Martinez et al. (2021) focuses on big-data services application to the power distribution sector. Khan et al. (2020) reviews autonomous maintenance via application of digital twin technologies. Mattioli et al. (2020) and Fink et al. (2020) review the field of general AI applications to improve production maintenance. Anomaly detection forms one aspect of predictive maintenance and Erhan et al. (2021) covers only this as applied to sensor-based systems. All the aforementioned reviews cover RL as *one* of the techniques and thus also include other ML techniques.

Panzer and Bender (2021) is a high-quality review of RL in production systems covering assembly, process and quality control, maintenance, logistics and energy management. However, only 8 articles (of 120) cover maintenance (and not necessarily predictive techniques). On the other hand (Zonta et al. 2020) is focused on predictive maintenance and recent too, which however, interestingly, does *not* cover RL at all.

To the best of our knowledge, this is the first systematic review of RL applied to *all* aspects of predictive maintenance.

Figure 2 shows the organization of the paper. After outlining our survey methodology in Sect. 3. Section 4 shows our innovative use of $tf - idf$ for mining techniques and industrial applications from the vast number of articles surveyed. The term "predictive maintenance" is a highly "all-encompassing" term and it is hard to find a standard definition—in Sect. 5 we describe the various maintenance strategies. A taxonomy for RL and PdM is designed in Sect. 8, while the remaining sections are devoted to a thematic survey and identifying challenges and suggesting future avenues for research. Finally the summarized insights as well as a practitioner's implementation guide of RL for PdM is provided in Sect. 14.

[2] As of July-2022, combined over Scopus, Web of Science and IEEE Xplore.

Table 1 Comparison of similar survey and review articles

Survey paper	PdM focused?	RL focused?[a]	RL articles	RL performance comparisons?	RL taxonomy presented?	Case-studies?	Mathematical treatment?	Focus/other notes
Panzer and Bender (2021)	All operations	RL only	8	✓	x	x	x	Production systems
Erhan et al. (2021)	All operations	ML + RL	5	x	✓	x	x	Anomaly detection only. High-level taxonomy
Barja-Martinez et al. (2021)	All operations	ML + RL	1	x	x	x	x	Big-data services for power distribution
Ren (2021)	PdM	ML + RL	5	x	x	x	✓	Light mathematical treatment
Khan et al. (2020)	PdM	ML + RL	2	x	x	x	x	Focused on digital twin technologies
Mattioli et al. (2020)	PdM	ML + RL	1	x	x	x	x	Symbolic AI combined with ML
Fink et al. (2020)	PdM	ML + RL	2	x	x	x	x	Industrial PHM
Zonta et al. (2020)	PdM	ML only	0	x	✓	x	x	Exhaustive taxonomy. Covers ML techniques only
This survey	PdM	RL only	108	✓	✓	✓	✓	Focused, exhaustive coverage

"ML" includes conventional ML and deep learning and "RL" includes deep RL

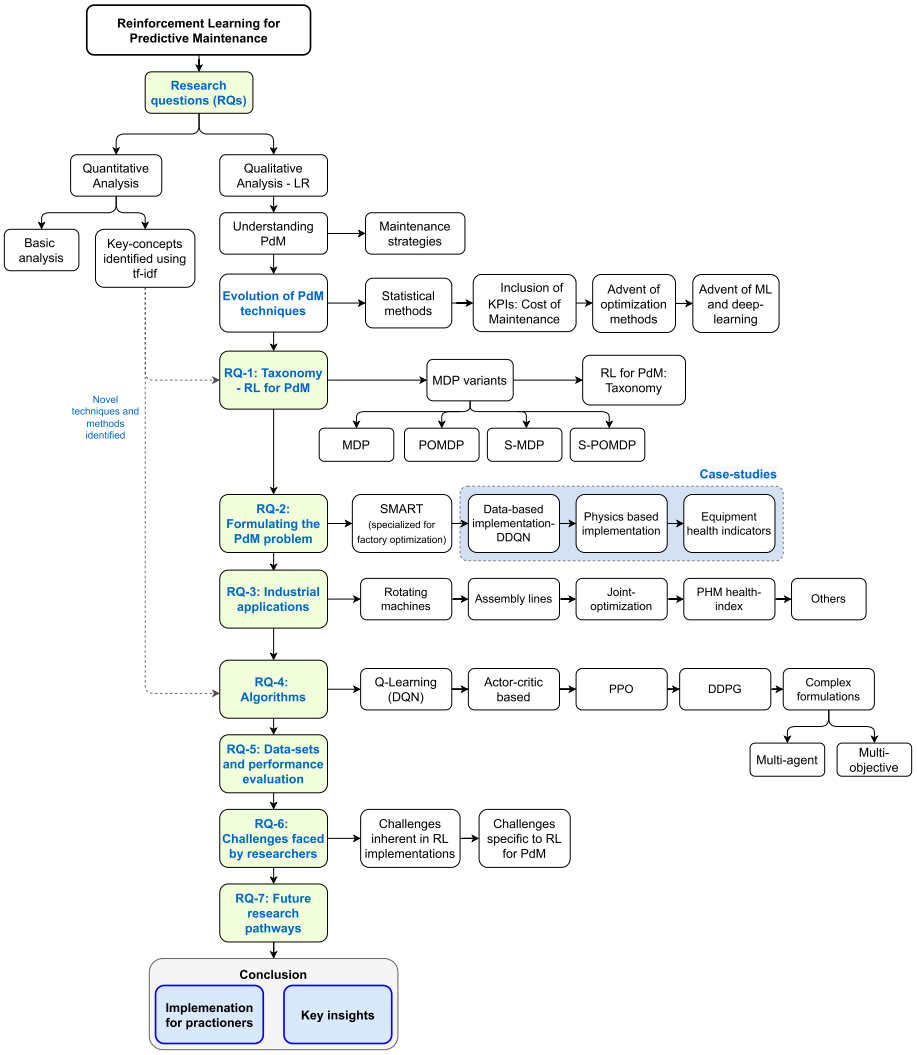


Fig. 2 Organization of the paper

3 Research methodology

In this section we lay out the research objectives (Table 2), and outline the research process. We adopt the six-step scientific and systematic methodology suggested by Templier and Paré (2015) and outlined in Fig. 3.

3.1 Research questions

Procedure for conducting the literature review

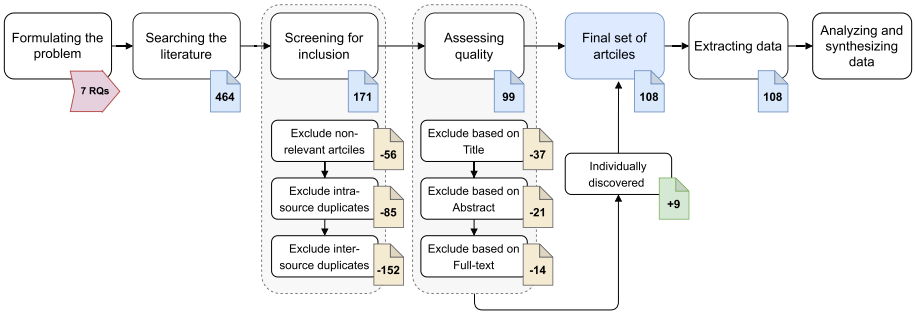


Fig. 3 Literature search process (Templier and Paré 2015), with number of articles at each stage

Table 2 Research questions

RQ	Research questions	Section
RQ-1	Taxonomy of RL methods for PdM	Sect. 8
RQ-2	How are PdM problems formulated as an RL task?	Sect. 9
RQ-3	What industrial applications has RL for PdM been explored for?	Sect. 10 (Table 12)
RQ-4	What RL algorithms have been successfully used for PdM?	Sect. 11 (Table 13)
RQ-5	How is the performance of the RL agent evaluated? What data-sets and metrics were used for evaluation?	Sect. 11 (Table 16)
RQ-6	What challenges have researchers documented in the application of RL for PdM?	Sect. 12
RQ-7	What are the future directions suggested by researchers?	Sect. 13

Step 1 *Formulating the problem* The primary aim of this research is a survey of how RL has been applied for predictive maintenance.

Inclusion criteria

1. *Diagnosis- and prognosis-term based inclusion criteria* To ensure that literature satisfying the various forms of PdM are not missed, we search for articles that cover all *three* aspects of machine "breakdown":

   (i) early prediction of anomalous behavior of equipment,
   (ii) early prediction of faults, and
   (iii) RUL prediction

2. *Maintenance strategy based inclusion criteria* We covered RL in the context of maintenance *strategies* falling under the broader umbrella of predictive maintenance:

   (i) Predictive maintenance
   (ii) Condition based maintenance (CBM)
   (iii) Prescriptive maintenance
   (iv) Preventive maintenance (PM)

(v) Prognostic Health Management (PHM)—While this is a strategy *beyond* predictive maintenance, RL techniques remain the *same* and hence included in scope.
(vi) RL applied for "scheduling" or "planning" maintenance.

3. *Time period inclusion criteria* Often review articles focus on current research (say last 5 years); we, on the other hand did not. This was to ensure that fundamentally solid research and implementations were studied. Two good examples of these are Das et al. (1999); Gosavi (2004b) published more than 15 years ago, but remain fundamentally valid and collectively hold more than 400 citations.

Exclusion criteria

1. Intra-index duplicates
2. Inter-index duplicates
3. Reviews, book chapters and conference proceedings

Step 2 *Searching the literature* Articles were searched using three primary journal index services: Scopus™, Web of Science™ (WoS) and the IEEE Xplore™, using a two-pronged approach i.e. individual search queries as well as an "all-inclusive" query.

1. *All-inclusive query* `'reinforcement learning' AND ('predictive' OR 'prescriptive' OR 'preventive' OR 'prognosis' OR 'prognostic') AND 'maintenance' AND ('fault' OR 'condition' OR 'anomaly')`

   Individual queries:
2. `'reinforcement learning' AND 'predictive maintenance'`
3. `'reinforcement learning' AND ('condition monitoring' OR 'condition based monitoring' OR 'CBM' ) AND 'maintenance'`
4. `'reinforcement learning' AND 'prescriptive maintenance'`
5. `'reinforcement learning' AND 'preventive maintenance'`
6. `'reinforcement learning' AND ('prognostic' OR 'prognostics' OR 'prognostic health management' OR 'PHM')`
7. `'reinforcement learning' AND ('anomaly detection' OR 'anomaly')`
8. `'reinforcement learning' AND 'fault' AND ('predictive' OR 'maintenance' OR 'prognosis' OR 'prognostic')`
9. `'reinforcement learning' AND ('RUL' OR 'remaining useful life')`

Step 3 *Screening for inclusion* Apply inclusion and exclusion criteria listed in Step-1 and filter articles for assessing quality.

Step 4 *Assessing quality* Titles and abstracts were studied to ensure articles were scope relevant—for example, articles related to data networks were filtered out. At times, full-text had to be studied.

Step 5 *Extracting data* From the full-text, we extracted and tabulated important information such as research novelty, the industrial application area, PdM KPI addressed, MDP/SMDP/POMDP formulation, RL algorithm, information related to agent type (single/

multi), state and reward formulations, evaluation data-sets used, performance metrics, research gaps and finally the future work identified by the researchers.

Step 6 *Analyzing and synthesizing data* Extracted information was organized and analyzed to identify patterns. It was mined for answering the Research Questions outlined above. Patterns, such as what was the most common algorithm used, what was the most common simulation or evaluation data-set used etc., were identified.

Step 7 *Including non-indexed articles* Occasionally we discovered an excellent article, that was not found in these three index data-bases. These were studied and included for their research contribution. There is no better example of a non-indexed article of high research contribution than Mahadevan et al. (1997). They outline their SMART algorithm for factory optimization problems (Sect. 9.2).

Table 3 shows number of articles at each stage against the search engine.

4 Quantitative analysis

Bibliometric tools provide quantitative analysis. Figure 4a displays an exponential rise in the articles over the last 5 years, faithfully reflecting IoT and sensor growth. Figure 4b shows expertise of authors in this field. We noticed that 4 of the top 5 authors have worked together on multiple papers.

We used the 'R' software[3] to create the "Keyword Co-occurrences" (Fig. 5a) and the "Conceptual Map" (Fig. 5b).

It was observed that all key-term analysis tools highlight terms that occur most frequently. This includes terms such as "reinforcement learning", "predictive analytics" and "decision making" and therefore does not enable identification of unique approaches to solve the problem, unique application areas or any other uniquely different highlight of the research. To overcome this limitation we propose the use of a Natural Language Processing (NLP) technique—Term Frequency—Inverse Document Frequency ($tf - idf$).

4.1 Using $tf - idf$ to identify novel concepts

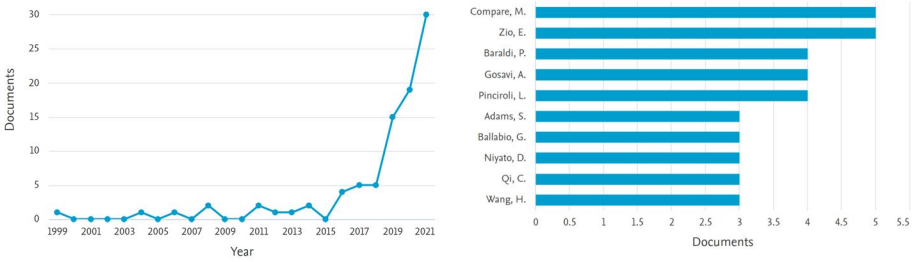
$tf - idf$ is a technique that vectorizes text (words converted to numeric values). The technique generates a score for each word that signifies its importance in the document and the entire corpus (refer to Appendix 3 for details).

We applied this on title, abstract and keywords of the articles and this produced a low score for the most-frequently occurring terms and enabled *unique* terms, with higher scores, to surface to the top. Scores ranged from 1.4054 to 6.4548 with the lowest score assigned to "reinforcement learning", the term that occurred in every document. The terms of interest will generally lie in the higher 1/3rd to 2/3rd range of scores. Figure 6 shows interesting techniques discovered by this method—"Petri Nets" (used to describe discrete-event distributed systems and which can be used as graph models for modeling the control behavior of systems), "particle filters", the Chapman-Kolmogorov equation (for relating

[3] The R Project for Statistical Computing: https://www.r-project.org/.

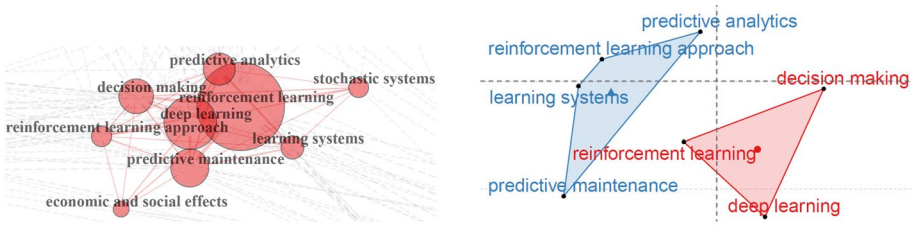
Table 3 Article mining: Scopus, Web Of Science and IEEE databases

	Raw articles	Reviews, books, etc.	Dups. intra-source	Dups. inter source	Filter based on titles	Filter based on abstract and full-text	Selected articles
Anomaly detection	32	11		9	8	1	3
Condition based maintenance	47	2	1	16	4	10	14
Fault diagnostics	120	24	12	33	16	20	15
Predictive maintenance	64	12	11	23	2	2	14
Prescriptive maintenance	5		2	3			
Preventive maintenance	83		40	1	5	1	36
Prognostic health management	34	2	11	5	2	1	13
Remaining useful life prediction	15	1	8	3			3
All inclusive query	64	4		59			1
Total	464	56	85	152	37	35	99
Individually discovered articles:							9
Grand total							108



(a) Articles by year (b) Contributing authors

Fig. 4 Analysis using Scopus™



(a) Keyword co-occurrences (b) Concept map

Fig. 5 Topic analysis using R Bibliometrix tool

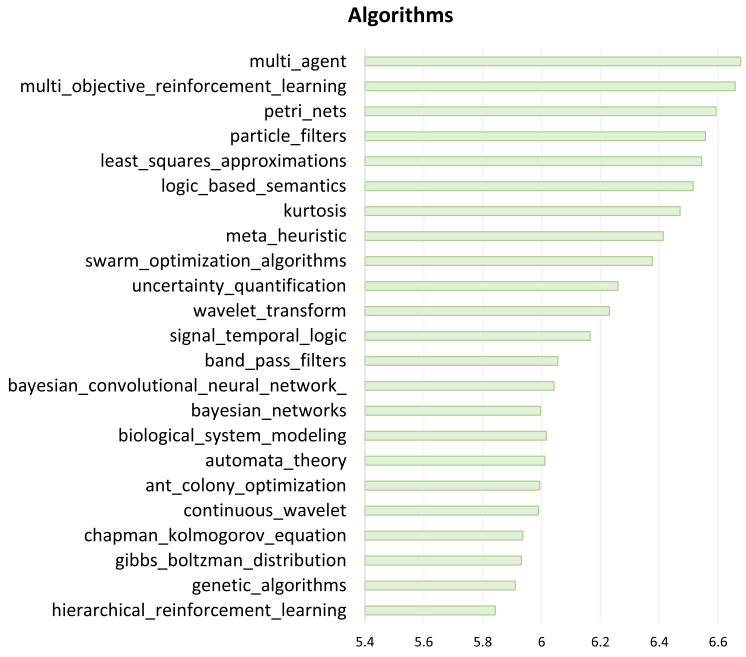


Fig. 6 tf-idf driven identification of important topics

joint probability distributions), multi-agent, ant- and swarm-optimization techniques etc. Table 14 lists these and other concepts found using this analysis.

5 Understanding predictive maintenance

To minimize downtime and maximize production levels and quality, manufacturers must make accurate and timely decisions. Maintaining manufacturing equipment has becoming challenging as there could exist a mix of older and worn-out machinery along side newer machines and increasing complexity of production processes due to demands of greater variability and customization. "Predictive maintenance" forms an important strategy adopted by organizations to address this challenge.

In literature, several definitions occur in the context of predictive maintenance. They present a mélange of terms with overlapping meaning: early fault diagnosis and fault prediction, estimation of RUL, and maintenance *strategies* such as conditioning monitoring, predictive and preventive maintenance, and prognostic health management. Organizations apply one or a mix of these maintenance *strategies*. To better understand them we present their definitions as stated by the National Institute of Standards and Technology (NIST) and their programs.

- *Reactive or corrective maintenance (CM)*: Maintenance is performed only after a machine fails or stops operating (within the specifications limits) (Thomas and Weiss 2020).
- *Preventive maintenance (PM)*: Weiss et al. (2016); Thomas (2020) define this as a time- or usage-based strategy. Pre-defined maintenance is performed after pre-defined intervals (τ), such as N operation-cycles, X hours etc. Correa and Guzman (2020) suggest that in addition to manufacturer recommendations; these intervals be scientifically based on statistical analysis of failure frequencies.[4]
- *Predictive maintenance (PdM)*: Thomas and Weiss (2020) define this as a strategy that measures the condition of equipment and gauges its reliability. Thus this strategy covers *reliability-centered maintenance* (RCM) and *condition-based maintenance* (CBM). Correa and Guzman (2020) state that PdM initiatives assisted the evolution of conditioning monitoring systems. Thomas and Weiss (2020) defines predictive maintenance as being analogous to condition-based maintenance. It is therefore important to understand CBM and its relation to Predictive Maintenance (PdM).

CBM requires condition monitoring systems (CMS) that periodically monitor the health and operating conditions of the machines with frequencies that are application specific—from milliseconds to minutes to hours. Measurements are enabled by sensors and embedded processors. It is possible to measure the condition and reliability at physical levels such as tool, component, machine and even assembly level; as well as at a higher *functional* level of the manufacturing process.

The concept of "prediction" indicates the use of current and historical measurements, provided by the CMS, and the use of analytical, statistical and machine-learning based

[4] A variant of PM is "opportunistic" maintenance; triggered when a machine fails before τ and is administered CM and all *other* machines receive PM.

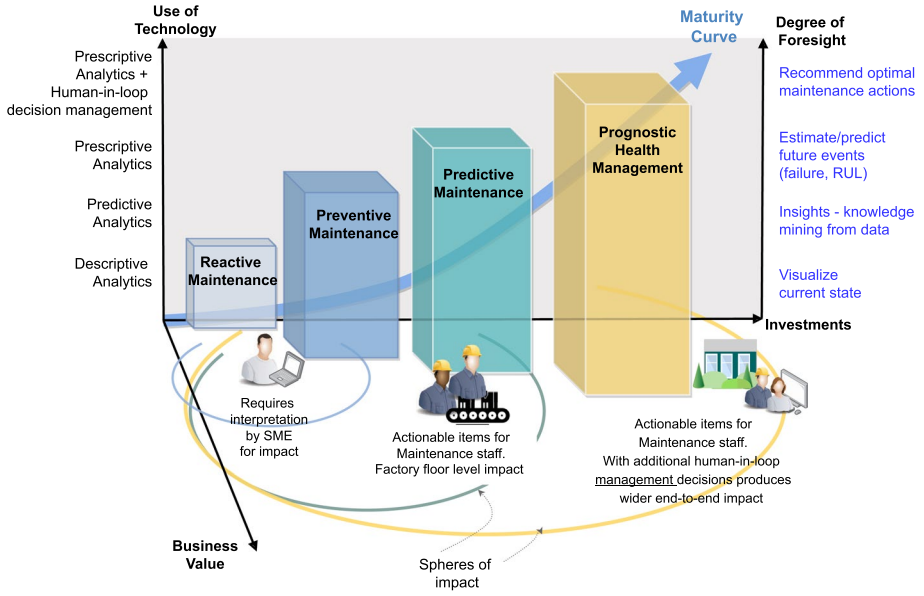


Fig. 7 The industrial maintenance prism

models to perform diagnosis and prognosis. **Diagnosis** is the science of determining *when* and *why* performance surpassed lower or higher production thresholds; and **prognostics** *estimates* when and determines whether they *will* be exceeded in the future, Vogl and Qiao (2021). Both these together allows estimation of the remaining-useful-life (RUL) of equipment.

PdM therefore assists in the efficient scheduling of maintenance and repairs and therefore prevent unexpected failures.

- *Prognostics and Health Management (PHM)* Evolution of maintenance science has led to prognostics and health management (PHM) which Weiss et al. (2017) defines as a combination of advanced monitoring, diagnostic, and prognostic technologies and the management around it, for minimizing the occurrence of failures. PHM goes beyond predictive maintenance to include *human expertise* for *reasoning* about what went wrong and the *management* of the related systems (staff, machines, planning, logistics etc.).

With the foundations laid down, we can now define predictive maintenance as:

> *Predictive maintenance* is an industrial science that utilizes condition-based monitoring technology to observe the health of machines, thereby enabling early detection of its deterioration via anomalies or faults; and provides a mechanism to plan and schedule maintenance actions to maximize its remaining useful-life.

The graphic in Fig. 7 shows PdM in the context of multiple perspectives: investment, business value, maturity of analytics, the impact on "end-users" and probably the most important—the level of foresight the strategy provides. PdM is uniquely placed and

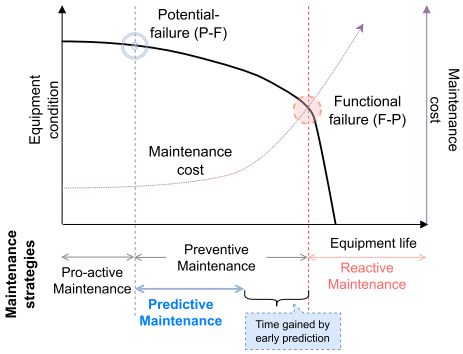


Fig. 8 The P–F curve

provides a descriptive view of the current state of machines all the way up to being able to recommend the best maintenance actions.

5.1 The P–F curve

The wear process of machines can be illustrated using P–F curves (Fig. 8). Potential failure (P–F) indicates the *earliest* "failure" state that can be detected. It is the point at which equipment begins to degrade. Functional failure point (F–P) is a state when the equipment has *reached* a failure state and is no longer operational. The interval between points P–F and F–P indicates the duration between detection of potential failure and the point the asset reaches a failed state. This duration is largely determined by the technology used to detect failure.

The various maintenance strategies are seen in relation to the P–F curve and a machine's useful life. Predictive maintenance enables an organization to lengthen this duration.

5.2 Business benefits of implementing PdM

Effective implementation of predictive maintenance drastically reduces the chances of failure events, thus improving productivity and directly reducing scheduled maintenance costs. Indirect benefits include significantly improved health, safety and environment compliance. Saving mechanisms and some estimates are provided next.

Potential benefits of implementing PdM, according to Coleman et al. (2017), include 5–10% savings related to maintenance, repair and operations and hence reduced inventory costs; 10–20% equipment up-time and availability; 20–50% reduction in time spent on maintenance planning; and 5–10% reduction in overall maintenance costs. Abudali and Siegel (2021) report a 5–10% gain in overall equipment effectiveness; 1–5% reduction in unplanned downtime; 10–20% in spare-parts reduction and an overall annual savings of $500,000 savings per plant. And for a specialized industry such as mining, a single field failure of a truck costs about $5,000 every hour, Zhang et al. (2019).

Predictive maintenance KPIs for optimal field service Business success of implementing predictive maintenance can be measured using suitable key performance indicators (KPIs) such as the Mean Time Between Failures (MTBF) that serves as an indicator of system-reliability; Mean Time to Repair (MTTR) that provides an indication of system availability, Mean Down Time (MDT), that is additive to MTTR and

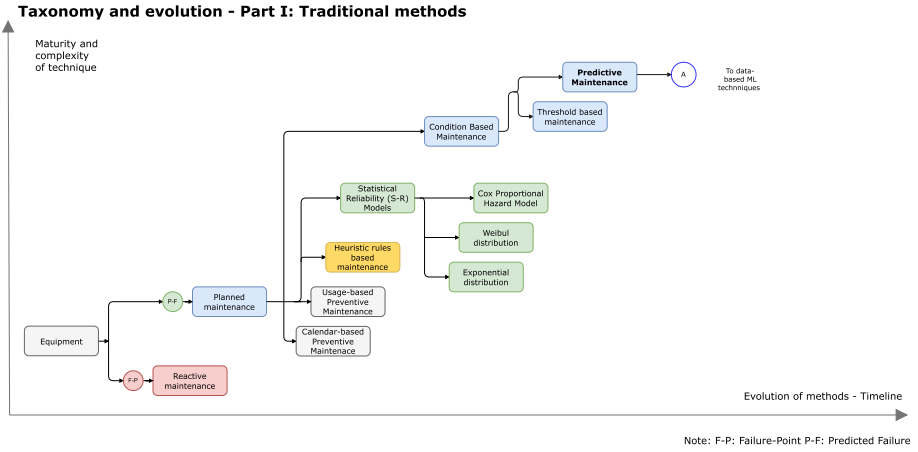


Fig. 9 Taxonomy and evolution Part I: traditional methods for PdM

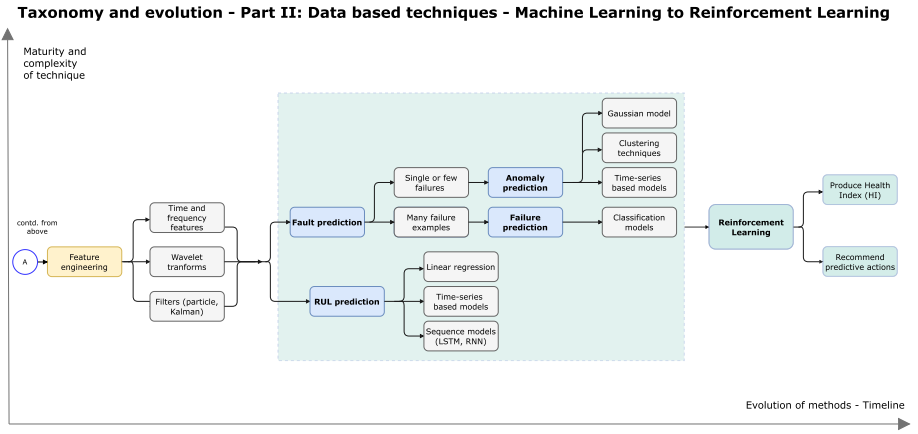


Fig. 10 Taxonomy and evolution Part II: data based (ML and RL methods for PdM

includes logistic delays related to replacement of parts and unsuccessful attempts during unplanned maintenance and finally; Overall Equipment Effectiveness (OEE) that describes overall equipment productivity.

6 Evolution of predictive maintenance methods

Predictive maintenance methods have matured and evolved over time—from reactive maintenance to planned maintenance using scientific techniques. Figures 9 and 10 show the evolution (in two parts), with the vertical axis representing maturity and complexity of the techniques. A taxonomy of RL specific techniques are detailed in Fig. 14.

6.1 Origins: statistical methods for reliability

Evolution of scientific "predictive" methods, Fig. 9, started with statistical methods for reliability (S–R). Understanding the foundations of these analytical methods, based on principles of physics, will help appreciate the benefit of RL methods.

Fundamentally, S–uses a continuous variable representing the lifetime of the equipment. This variable follows a probability distribution function (PDF) that helps determine the probability of failure thereby providing a trigger to perform preventive maintenance. Common PDFs representing failure times are the exponential, normal, log-normal and the classic, widely used, Weibull distribution (Mann et al. 1995).

We first look at the simplest form of S–R based PdM decision making. The Weibull model is a parametric model where the reliability function is given by (1), where $R(t)$ is the probability of survival from $t = 0$ to time t and β the scale, and α the shape parameter.

$$R(t) = e^{-(\frac{t}{\beta})^{\alpha}} \tag{1}$$

In E. H. Wallodi Weibull's original 1951 paper (Weibull 1951), several wide ranging applications are shown—for example the fatigue life of steel, while Abernethy (2018) considers this model as the world's most popular statistical model for life data.

Mann et al. (1995) explains how the Weibull distribution helps plan maintenance: When α is less than one (decreasing failure rate) or equal to one (constant failure rate), equipment maintenance may not be cost-optimal and when α is greater than one (increasing failure rate), one can directly read the maintenance time interval from the probability plot, for an acceptable failure probability. Bala et al. (2018) used a three-parameter Weibull distribution to model data of load haulers used in underground mines. Preventive maintenance intervals were identified based on the *subsystem* level reliability estimates of shovel dumpers, dragline excavators and in-pit crushers.

Distributions have been used by RL researchers in solving PdM. Gosavi (2004b) uses the *gamma* distribution for time between machine failures and that taken for machine repair. Gamma is also used for modeling production time, while the production demand is Poisson. The time taken for preventive maintenance is a uniform distribution.

6.2 Inclusion of KPIs: total maintenance cost objectives

Business KPI based objectives evolved next. We take a case with a *cost* based PdM KPI, that assumes a component that follows the Weibull wear distribution (1). As is normally the case, cost of unplanned failure C_{CM}, is greater than cost of planned preventive maintenance C_{PM}. Then, there exists an optimal replacement interval exists that can be found by minimizing the cost per unit time, $TC(t)$ given by (2) (Mann et al. 1995):

$$TC(t) = \frac{C_{PM} \cdot R(t) + C_{CM} \cdot (1 - R(t))}{\int_0^t e^{-(\frac{x}{\beta})^{\alpha}} dx}, \tag{2}$$

where $R(t)$ is the probability of survival (reliability parameter) from (1) and x is the Weibull random variable.

6.3 Advent of optimization methods

The next evolution of methods added *schedule optimization* of maintenance, using numerical techniques such as linear and nonlinear programming. Real-world optimization problems involve multiple conflicting objectives (for example cost and frequency) that must be considered simultaneously. These are handled using vector-optimization formulations and involves a three-pronged approach: decision-making methods, mechanisms for solving nonlinear constraints and the core optimization algorithm to minimize the objective function, Russenschuck (1999).

Li et al. (2016) demonstrates the use of mixed-integer programming (MIP) to solve the maintenance scheduling for a fleet of aircraft. They use AMPL, the modeling language for CPLEX which is a widely used industrial-grade optimization software for linear and integer programming. CPLEX uses the "simplex" algorithm, a linear programming method.

6.4 Advent of ML and deep learning techniques

The next phase of evolution involved ML followed by deep learning (DL). Many excellent review articles (Fink et al. 2020; Mattioli et al. 2020; Ren 2021) have covered ML algorithms employed for predictive maintenance.

Eke et al. (2017) predict transformers failures using classic unsupervised-learning techniques such as PCA (for pre-processing) and k-means clustering, to analyze the concentration of dissolved gas in oil with-respect-to the different operating periods. Scheibelhofer et al. (2012) apply used supervised-learning techniques—CART decision trees and Random Forests to optimally predict maintenance schedules in the quality sensitive semi-conductor industry. Failures of ion implantation tools are predicted and the use of decision trees allow human understandable failure modes. Kabir et al. (2018) apply Random Forest and AdaBoost to predict which transformers are most likely to fail.

Finally, we look at a carefully chosen deep learning article that carries a connection to RL via the "state-goal" concept. Skydt et al. (2021) applied LSTM (Long Short Term Memory) techniques using voltage, active power, and current as input features. The authors used "state-goals" and not raw features to provide a sort of normalized form of point data. Equation (3) is an example of a state-goal representing voltage. It includes a "look-back" parameter that in turn determines $V_{\max}$ and $V_{\min}$. The look-back parameter affects sensitivity—too much historical data leads to state-goals that do not allow for predicting failure; while including too little historical data can lead to state-goals that do not carry necessary information and operational variation for prediction.

$$V_g = \begin{cases} 1 - \frac{|V-\bar{V}|}{V_{max}-V_{min}}, & \text{if } V_{\max} \neq V_{\min} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Having looked at the evolution PdM methods from statistical to deep learning we are now ready to understand RL and apply it PdM.

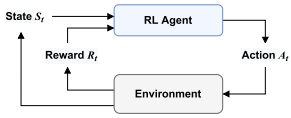


Fig. 11 The basic RL flow (Sutton and Barto 2018)

7 Understanding reinforcement learning

Often real systems are highly complex and have non-linear dynamics. Thus it is hard to create a model of the system to apply analytical methods. In this section we briefly introduce RL concepts and in Sect. 9 we will formulate the PdM as an RL task.

ML is broadly divided into supervised and unsupervised methods. Supervised used "labeled" data to train a "learner" and predict on newly presented data or classify them under one of the learnt labels. Unsupervised methods learn patterns from "un-labeled" data.

RL is based on learning methods of animals and humans. Learning takes place by interacting with the environment. They learn from the consequences of actions performed, rather than from explicit supervised teaching and therefore does not depend on structured data (labeled or unlabeled). RL is therefore a more natural learning process based on "trial and error".

RL is essentially a feedback process loop as seen in Fig. 11. The agent acts by selecting actions and interacts continually with its environment. The environment offers quantified feedback on how good or bad the action performed was, using a scalar value termed as *rewards*. RL algorithms are thus methods designed to train the agent to maximize the rewards and *reinforcing* the good actions over bad. Overtime the agent learns an optimal *policy* of making the right decisions.

In the case of PdM the agent is the "planner" and our goal is to train an optimal planner. The environment consists of everything outside the planner: sensors attached to the machine, job load, operators, dust/humidity etc. It can also include information from other systems that one plans to use to train the agent, for example historical maintenance records.

7.1 Mathematical framework

This optional sectional describes the basic mathematical framework ending with the central Q-Learning algorithm equation.

Reinforcement Learning is formalized using the mathematical framework of Markov Decision Processes (MDPs), (Achiam 2018b; Icarte et al. 2022; Sutton and Barto 2018). MDPs are defined by a 5-tuple $\langle S, A, R, P, \rho_0 \rangle$, where S is set of all valid states, the environment can exist in; A, a set of all actions; $r : S \times A \times S \to \mathbb{R}$ is the reward function,[5] with $r = R(s, a, s')$. P: $S \times A \to \mathcal{P}(S)$ is the transition probability function, where $P(s' \mid s, a)$ provides the probability of the current state s transitioning into s' on taking the action a. Finally, ρ_0 is a distribution of initial state. Note that we could use time index and denote s as s_t and s' as $s_{(t+1)}$.

The **policy**, is a *mapping* of what action to take, given a state, to produce the best rewards. A stochastic policy is denoted by π, with learnable parameters θ:

[5] For certain RL problems, for example when the reward is stochastic, this definition may not strictly apply. For the problem to be solvable, it is sufficient if the expected value is a function of s, a, and s'.

$$a_t \sim \pi_\theta(\cdot p \mid s_t) \tag{4}$$

τ is a trajectory (5) representing a sequence of states and actions and Markovian state transitions (6), caused by an action, which in turn was suggested by a policy.

$$\tau = (s_0, a_0, s_1, a_1, ...) \tag{5}$$

$$s_{t+1} \sim P(\cdot \mid s_t, a_t) \tag{6}$$

Rewards are computed using a reward function R, that depends on the current state s_t, the action performed a_t, and the next state s_{t+1}. The time-indexed r_t is computed using (7); often simplified to depend only on the current state $r_t = R(s_t)$, or the state-action pair $r_t = R(s_t, a_t)$.

$$r_t = R(s_t, a_t, s_{t+1}) \tag{7}$$

Returns given by (8), provide a long-term measure of reward. **Discount rate** γ $(0 \leq \gamma \leq 1)$, allows one to control the impact of selecting an action; for $\gamma = 0$, the agent only learns those actions that produce immediate rewards, whereas values closer to 1.0 incorporate distant future impact.

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t, \tag{8}$$

Value-functions provide a way of knowing the *value* of being in a state. Two common forms exist. A state based form (9), denoted by $V^\pi(s)$, is the expected return starting in state s and governed by policy π; and a state-action form (10), denoted by $Q^\pi(s, a)$, that provides the expected return if action a is performed while in state s.

$$V^\pi(s) = \mathop{\mathbb{E}}_{\tau \sim \pi} [R(\tau) \mid s_0 = s] \tag{9}$$

$$Q^\pi(s, a) = \mathop{\mathbb{E}}_{\tau \sim \pi} [R(\tau) \mid s_0 = s, a_0 = a] \tag{10}$$

7.2 Bellman equation and Q-learning

In general, the objective of a reinforcement learning task is to seek a policy that gathers the maximum reward, over the long run. The Bellman equations represents the value function as a sum of two parts—the immediate reward and the discounted future values.

The Bellman equation is an optimality condition describing the optimal action-value function, $Q^*(s, a)$ (11).

$$Q^*(s, a) = r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim P} \left[\max_{a'} Q^*(s', a') \right] \tag{11}$$

Many RL approaches use this recursive relationship.[6]Lillicrap et al. (2015); Achiam (2018a) describes how the Bellman equation is used for the Q-learning in the DDPG (Deep Deterministic Policy Gradient) algorithm. Equation (11) provides a formulation to iteratively learn $Q^*(s,a)$ or apply function approximators, such as neural-networks, where the parametric *loss function* can be designed using the mean-squared Bellman error function L (12) (Achiam 2018a).[7]

$$L(\theta, \mathcal{T}) = \mathop{\mathbb{E}}_{(s,a,r,s',d)\sim\mathcal{D}} \left[\left(Q_\theta(s,a) - \left(r + \gamma(1-d) \max_{a'} Q_\theta(s',a') \right) \right)^2 \right], \tag{12}$$

where $\mathcal{D}$ is the set of transitions, *max* finds the "optimal" action and d is a boolean variable with `True` representing end of an episode.

8 Taxonomy of RL for predictive maintenance (RQ-1)

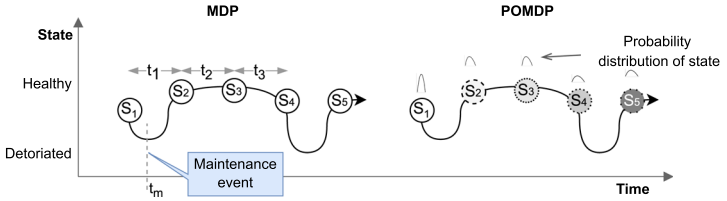
We now develop a taxonomy for RL for predictive maintenance. RL algorithms are classified as being *model-based* or *model-free*. A "model" in MDP/RL literature refers to a "mapping" or a "look-up table" consisting of probabilities. Learning the model essentially means learning the probabilities of state transition and the dynamics of rewards for a given action and state combination. This helps create the internal MDP and once built, it enables the agent to plan what it should do next as it is able to predict the reward for some action before executing it. **Model-free**, on the other hand, interact with the environment and *learn the consequences of the actions by trial-and-error*, Sutton and Barto (2018).

The classic MDP assumes a "completely observable" environment i.e. the states are fully measurable and the environment is stationary (the transition function does not change over time). Predictive maintenance is an extremely complex scenario and this assumption does not hold true. External situations such as production load variations, maintenance performed and/or missed, weather, power fluctuations can all severely affect the transition probabilities. PdM situations solved by multiple agents (multi-agent RL) is necessarily a non-stationary environment as an action by one agent changes the environment for another agent.

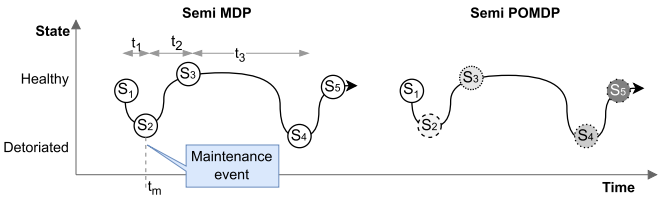
Therefore, realistic non-stationary environments are modeled using *multiple* MDPs that in turn model a small number of *non-observable* phases or modes of the environment. This method is known as **Hidden-Mode MDPs** (HM-MDPs), which in turn are a subclass of **Partially Observable MDPs** (POMDPs). **Partially Observable MDP** (POMDP) extend MDPs to partially observable environments, where the agent is not able to observe the current state directly but rather through noisy observations for e.g. using sensory measurements (Patil and Abbeel 2013). The actions are therefore based on historic information prior to the current time step i.e. previous states, observations and actions. The current state, called a *belief* state, is thus represented as a probability distribution over states. POMDPs tend to be better than HM-MDPs for representing non-stationary systems, while

[6] Not all RL algorithms are value based; some estimate the return over long trajectories governed by policies (Swazinna et al. 2022; Schaefer et al. 2007). However, in general, they all seek the Bellman optimality condition.

[7] The Bellman equation is designed to consider *all* state-action pairs. In case of partial, sampled data, using it as a surrogate objective for value prediction, is a poor choice (Fujimoto et al. 2022).



(a) Fixed/discrete time variants: MDP (fully-observable states) and POMDP (partially-observable states)



(b) Variable/continuous time variants: Semi MDP (for fully-observable states) and Semi POMDP (for partially-observable states)

Fig. 12 MDP variants

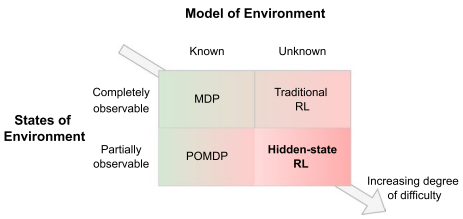


Fig. 13 Four categories of RL-PdM problems. Degree of difficulty increases top-left to bottom right (green to red). (Color figure online)

HM-MDPs provide a more simpler mechanism to formulate non-stationary problems since they require fewer parameters.

MDP, POMDP and HM-MDP are all fixed interval discrete time formulations. In **Semi-Markov decision processes** (SMDPs) the actions are permitted to take *continuous* and *variable* length of time. Transition from current state to next depends on the action taken as well as the time elapsed since the action was taken—called "sojourn" time spent in the previous state. Figure 12a shows how having equal time intervals, $t_1 = t_2 = t_3$, misses the maintenance event at t_m. In Semi MDP (Fig. 12b), t_1 represents the sojourn time the system spends in state s_1, before maintenance takes place at time t_m, which converts the system into state s_2 and spends t_2 in this state before transforming into a healthier state s_3. As seen in the figure $t_1 = t_m$ and $t_1 \neq t_2 \neq t_3$. Semi HM-MDP take this one step further to allow representation of realistic non-stationary industrial systems. This is all very pertinent to modeling the environment for predictive maintenance, if one decides to apply *model-based* RL methods. These concepts lead us to Fig 13 that shows the difficulty of solving RL-PdM problems based on the above concepts.

RL algorithms can also be classified as value iteration, policy iteration, and policy search algorithms (Busoniu et al. 2017). **Value iteration algorithms** seek the optimal

value function, which in turn produces the optimal policy. The optimal value function represents the maximum return from every state or from every state-action pair. **Policy iteration algorithms** evaluate policies by constructing their value functions (non-optimal), these are then used to further find new, improved policies. The third sub-class, **Policy search algorithms**, directly learn what is essentially of interest i.e. the optimal policy.

A special class of methods known as Actor-Critic (A-C) methods are dual network, with the policy structure, called "actor" independent of the estimated value function, the "critic". A-C posses three main advantages—firstly, they learn the optimal probabilities of selecting various actions and this is a significant advantage for non-Markovian environments such as POMDP (Jaakkola et al. 1994; Singh et al. 1994); secondly, they require lower compute to select actions and are especially suited for continuous-valued actions (infinite action space); and thirdly, keeping the actor separate enables addition of domain specific knowledge on the set of allowed policies—and this should assist incorporation of predictive maintenance domain knowledge (Sutton and Barto 2018).

9 Reinforcement learning for predictive maintenance (RQ-2)

In conventional operations research, the planning and scheduling of jobs or tasks (such as maintenance) have been solved by mathematical modeling. This is true if the problem can be simplified and properly formulated.

Predictive maintenance tasks are dynamic and complex in nature. Modeling this problem mathematically and then optimizing for factors which can include downtime costs, reliability targets, penalties by end-customers, labor costs etc. can all create a very complex problem to solve. On the other hand, a model-free RL approach will prove more effective.

9.1 Formulating the PdM problem

RL involves designing three essential components—an environment that will provide feedback for the agent to learn from, the list of actions that the agent will perform and the "reward" function.

PdM is an optimization problem. Performing maintenance too late, risks machine failure and loss of production. Perform it too early or too often results in unnecessary costs.

In PdM our objective is to determine if maintenance is due based on equipment condition. Essential machine health can often be determined by sensors that measure physical parameters such as vibration, temperature, tool condition, fluid pressure, lubricant condition etc. It is obvious that these parameters will vary as per the industry—for example a mining equipment is subject to very different conditions when compared to a textile machine. A PdM algorithm, at any given point in time, must take decisions based on the "health condition" of the equipment: perform maintenance, replace parts or do-nothing.

Figure 16 shows a typical and simplified state diagram. S_1 is the starting state. As time progresses the equipment is used and undergoes wear-and-tear as it moves from S_1 to S_2 to S_3 and beyond. No action is performed by the agent so far. At S_{t1}, a maintenance action is performed and the equipment can be thought of regaining "remaining-useful-life". We show this by a blue arrow and the equipment attains state S_3 and once again the equipment is used and its RUL reduces. We then show an advanced stage of wear-and-tear where a replacement of part is suggested at S_{t2} and implemented. This allows the equipment to go back to the initial state S_1. In case parts are not replaced the equipment progresses to a "failure" state, and the

state diagram reaches it's *terminal* state, S_T. While this is an over simplified view, it helps understand the MDP formalization, and the state and actions. Note that, a more accurate representation will, for example, have S_3 branch out all three action possibilities, with certain probabilities.

The terminal state S_T is important, in that it determines the type of RL learning task—and as such this is an *episodic* task as against a non-episodic continuous task.

Reward can be linked to a combination of PdM KPIs described in Sect. 5.2. It could be a measure of the impact of downtime caused by machine failure and could also involve the actual cost of replacement of the parts, labor costs etc. As an example Knowles et al. (2011) uses a combination of *profit*, repair cost and maintenance cost as a reward measure and use MTBF as a criteria to validate the performance of the agent.

We formulate PdM as an RL problem in Fig. 17. The blue text represents PdM specific features mapped to the standard RL elements.

9.2 The SMART family

Mahadevan et al. (1997) developed SMART, an early and significant algorithm for predictive maintenance. It is a model-free, "average reward" optimizing algorithm suitable for continuous time semi-MDPs (SMDP), suitable for several factory optimization problems. SMDP can model a continually changing state. The decisions however are discrete-time events. The reward formulation is *unique*—an immediate scalar reward and an accumulated reward at a fixed-rate until the next decision event. Machine failure is modeled as a gamma distribution. It can handle large state spaces (10^{15} states). We very briefly describe the core of the SMART algorithm and then summarize the evolution of its variants in Table 5. Gosavi, an expert in the field of applying MDP and Reinforcement Learning to operations management, additionally formulated the Q–P algorithm (Gosavi 2004a).

Equation (13), average sum of rewards, based on the Bellman equation, is the core step in the SMART algorithm (Mahadevan et al. 1997).

$$Q_{n+1}(s,a) \xleftarrow{\alpha_n} \left(r_{imm} - \rho_n \tau + \max_{a'} Q_n(s',a') \right) \tag{13}$$

$$\rho = \frac{R - C_{CM} - C_{PM}}{T} \tag{14}$$

where

$\xleftarrow{\alpha_n}$ abbreviated notation for stochastic approximation update rule
r_{imm} is the accumulated cumulative reward (nth and $(n+1)$th decision epochs)
τ is the transition time, s', a' is the next state/action at next decision epoch
ρ is the reward rate, R is the production revenue, C_{CM} is the corrective maintenance cost, C_{PM} is the preventive maintenance cost, T is the total time

9.3 Case-studies

We study three examples, each with a different aspect detailed technically. Collectively they should provide the reader with an overview of how the solutions are designed, assumptions made, implementation done and finally performance evaluated.

9.3.1 Data-based implementation: turbo-fan engine—deep double Q-network (DDQN)

Hoong Ong et al. (2020) detect anomalous behavior by analyzing an equipment's health. Equipment are embedded with sensors and an additional device that *aggregates* these raw signals. The RL agent learns an optimal maintenance policy and is able to recommend one of the following actions—repair equipment, replace equipment or no-action.

The PdM problem is cast as an MDP with *fully observable* states and defined by (15) where the state transition probabilities are given by the distribution P and γ is the discount factor. The state (16) is a simple binary number indicating the temperature range.

$$MDP = \langle S, A, R, P, \gamma \rangle \tag{15}$$

$$S^{\tau} = \begin{cases} 0, \text{ if } \tau \in [25, 60] \\ 1, \text{ if } \tau > 60 \end{cases} \tag{16}$$

The reward function R is designed to maximize the total productive up time of equipment and simultaneously adhere to budget constraints for maintenance events.

The RL algorithm employed is a Double Deep Q-Network (DDQN) (Hasselt et al. 2016) and is evaluated on the NASA C-MAPPS data set.

The novelty of research lies in sampling of the experience replay buffer. Hoong Ong et al. (2020) state that normal ϵ-greedy approach has a bias toward sampling experiences with high reward repeatedly. This is inefficient for a sparse-reward problem which is typical of a predictive maintenance problem. In addition to sparse rewards, the RUL/time-to-failure cycles for each equipment are different. They propose a Prioritized Experience Replay (PER) approach to compliment DDQN. The agent's ability to learn an optimal policy was significantly improved by the PER approach and produced a cumulative mean reward of 95.0, in 1.2×10^4 time-steps.

9.3.2 Physics-based implementation: rolling bearings—auto-step reinforcement learning

In this case study we see how a physics based formulation is implemented by Afshari et al. (2014). They apply the Paris Law along with Auto-step RL (AS-RL) (Mahmood et al. 2012).

A vast majority of industrial equipment use rolling bearings that are subject to frequent failure. Paris Law (Paris-Erdogan equation) is the best known model for fatigue crack growth over time, from the threshold zone to final failure.

$$\frac{dl}{dN} = C_0(\Delta K)^{\eta}, \tag{17}$$

where l is the length of a dominant crack, N is the number of usage cycles, ΔK is the stress intensity expressed as a range, C_0 and η are material constants.

A model for crack growth, described by its defect area D, is given by:

$$D = C_0(D)^{\eta}, \tag{18}$$

The crack growth model is first simplified by integrating (18):

$$ln(D) = \alpha + \beta\ ln(t + t_0), \tag{19}$$

where t is the time related to the P–F (Potential Failure Point), i.e. the earliest time when the smallest defect area can be detected.

AS-RL was used to learn the optimal parameters of the Paris formula $\theta(t) = [\alpha\ \beta\ t_0]^T$. Real experimental data was *not* available and the "actual" data was *simulated* using equation (19) with a little Gaussian noise added. The AS-RL is compared with a nonlinear recursive least square (RLS) adaptive filter algorithm using the root-mean-squared-error (RMSE) and standard-deviation σ metrics. Results are impressive and indicate that AS-RL has a RMSE of 0.20827 mm^2 compared to 8.0258 mm^2 of RLS and σ of 0.11037 mm^2 and 7.0035 mm^2, respectively.

Predicting RUL requires that a failure-threshold value be selected based on the length of crack on bearing. RUL is then computed as the time between the *current* state and the threshold. With a threshold of 25 mm^2 a RUL of 9.58×10^6 cycles was predicted.

Walsh (2022) cautions that Paris equation gives good results only for *long* cracks and when the material constants are known.

9.3.3 Equipment health indicators: turbofan degradation—TD learning with function approximation

While RUL provides an estimate of days to failure, it does not provide information on when to plan maintenance or the degree of maintenance required. New breed predictive maintenance systems are able to estimate the future "health" of the equipment, by computing a "health-index" $H(t)$ as a function of time t.

In Zhang et al. (2019) they term this as Health Indicator Learning (HIL) and suggest formulating it as a *credit assignment* problem. They assume the health degradation process as an MDP, with the value function v under policy π, given by the Bellman equation (11), redefined as (20).

$$V^{\pi}(s) = \underset{\tau\sim\pi}{\mathbb{E}}[(R_{(t+1)} \mid S_t = s) + \gamma V^{\pi}(S_{(t+1)} \mid S_t = s)] \tag{20}$$

Environment feedback via immediate reward R_t is measured under three PdM transition scenarios—normal to normal operation, normal to failure state and failure to failure state, (21).

$$R_t = \begin{cases} 0, & \text{if } U(s_t) = U(s_{t+1}) = 0 \\ -1.0, & \text{if } U(s_t) = 0, U(s_{t+1}) = 1 \\ R_{ff}, & \text{if } U(s_t) = 1 \end{cases}, \tag{21}$$

where $U(s) \in \{0, 1\}$ are labels, where '1' indicates failure (i.e. end of life for equipment) while '0' is non-failure i.e. normal-operation. R_{ff} is a hyper-parameter that needs to be tuned (but constrained to be $R_{ff} > -1.0$ for penalizing failure-to-failure transitions).

The model-free HITDFA algorithm, is a simple Temporal-Difference (TD) learning algorithm. The value function $V(s)$ is learnt by parameterizing it with θ, which could be a deep neural network. They implement an experience memory buffer and use *real* experiences instead of an expected value to update $V(s)$. This helps in removing correlations in the observations. Two hyper-parameters are learnt γ and R_{ff} by applying grid-search to ensure $H(t)$ meets the two constraints they impose (monotonic and minimum variance).

For evaluation they use the NASA Turbofan data set (Saxena and Goebel 2008). And since ground truth data is not available they assume the degradation model as exponential as per (Saxena et al. 2008) (22).

$$H(t) = 1 - D_0 - \mathrm{e}^{(at^b)}, \tag{22}$$

where, D_0 is the initial degradation state while a and b are wear-rate coefficients that depend on the effect of temperature, vibration and other system stress parameters.

When compared to other HIL methods, such as in (Li et al. 2013; Liu et al. 2013; Ramasso 2014), using MAPE as the metric, the performance is better by about 33.95%, on an average over four data-sets. An interesting outcome of the work is that the health indicators learned from their method can be regressed to predict RUL, albeit with a lower performance, measured using RMSE, when compared to five other methods, including SVR (Support Vector Regression) and CNN (Sateesh Babu et al. 2016) and LSTM (Zheng et al. 2017b). However, a health indicator retains the advantage of being a better indicator for predictive maintenance than a single RUL number.

Case studies provided a deep-dive into a few research articles to understand how RL was applied for PdM. We now conduct a wider literature review arranged by themes.

10 Industrial applications and use-cases (RQ-3)

10.1 Rotating machines, bearing housing, milling machines and hydraulic actuators

Several articles do *not* directly indicate a predictive maintenance action. We review some literature that focus on fault classification or diagnosis but present some sort of *predictive* machine health indication which can then be indirectly linked to providing maintenance suggestions.

Eltotongy et al. (2021) address predictive maintenance of machines with bearings using a novel combination of CNN and RL. They pre-process vibration time-series data using continuous wavelet transform (CWT) before feeding into a CNN. The CNN hyper-parameters are designed using a neural architecture search (NAS) approach based on RL. They evaluate the design on the Case Western Reserve University bearing data-set (Li 2019). Bearing faults are of three forms: inside-race, ball-fault and external-race. Severity is indicated by the diameter of the fault in 4 progressive stages: 0.1778, 0.3556, 0.5334 and 0.7112 mm. The fault classification accuracy reported is 99.34%. By treating the first stage of 0.1778 mm as P–F (Potential Fault) stage, we could initiate a maintenance action if detected. For the same data-set (Ding et al. 2019) used Q-Learning and compared it against SAE-Softmax (stacked autoencoder) method, see Table 6.

Cracking of gears is a frequent problem in heavily used rotating machines. Early detection of fault is addressed by Dai et al. (2020) using a Q-network and the reciprocal of smoothness index. They use RL for designing an optimal band-pass filter and find it

better than three methods: fast kurto-gram, Gini index-gram and Smoothness-gram. Cheng et al. (2018) model the degradation process of bearings using Calinski-Harabaz clustering index (Caliński and Harabasz 1974); use a state vector composed of six time domain features—mean value, RMSE, crest factor, average power, skewness and kurtosis and finally use Q-Learning and a reward function based again on the Calinski-Harabaz index to learn a PdM policy.

10.2 Assembly lines, multi-machine systems

Assembly and production lines comprising of multiple machines present highly complex dynamics (Li et al. 2009). Strategies applied are usually a combination of corrective maintenance (CM) and preventive maintenance (PM). Single machine solutions cannot be directly extended to multi-unit maintenance. Huang et al. (2019, 2020), Ling et al. (2018), Valet et al. (2022), Wang et al. (2015, 2016), Zhang and Tang (2022), Zheng et al. (2017c), Zhang and Si (2020) have studied two or more machine systems, often with in-line buffers. Table 7 shows a comparative summary of their research.

A typical problem formulation for multiple-machines with buffers is found in Huang et al. (2020). System is modeled as an MDP. The state is defined by (23); composed of machine age—specified by the probability of random failures on each machine $g_i(t)$, buffer levels $b_i(t)$ and the remaining maintenance duration $d_i^r(t)$. Actions a_t are binary (24), and Reward by (25), where c_i^{CM} and c_i^{PM} are the costs of resources incurred by CM and PM respectively. $c_p \cdot PL(t)$ represents the loss of profit due to break in production.

$$\text{State } s_t = [\, g_1(t), \ldots, g_M(t),\ b_2(t), \ldots, b_M(t),\ d_1^r(t), \ldots, d_M^r(t)\,] \tag{23}$$

$$\text{Action } a_i(t) = \begin{cases} 0, \text{ leave machine } M_i \text{ as is} \\ 1, \text{ turn off machine } M_i \text{ for PM} \end{cases} \tag{24}$$

$$\text{Reward } R(t) = -c_p \cdot PL(t) - \sum_{i=1}^{M} w_i(t) c_i^{CM} - \sum_{i=1}^{M} a_i(t) c_i^{PM}, \tag{25}$$

Effective PdM systems should not only be able to predict incipient faults, they must also be designed to handle faults that they fail to predict. When such a fault does occur, the system needs to respond as quickly as possible. This is the premise of Liu et al. (2017) who present a fault-tolerant control system for non-linear, multiple-input multiple-output (MIMO) system by devising an actor-critic methodology and a novel long-term cost function. A reduced computational burden is achieved by updating the Euclidean norms of weights of the neural-networks, instead of the actual weights. They evaluate the model on a *simulated* MIMO system and simulated incipient fault time-profiles Φ given by (26), where t_f is the time the unknown fault occurs.

$$\Phi(t - t_f) = \begin{cases} 0, & \text{if } t < t_f \\ 1 - e^{-0.0075(t - t_f)}, & \text{if } t \geq t_f \end{cases} \tag{26}$$

Actor network approximates the near optimal control policy. The proposed systems achieves 20% lower cost and a 54% reduced computation time, when compared to a neural network based system and even better performance when compared a linearly parameterized RL implementation: 55% lower cost and 77% reduced computation time. While a

separate stability analysis is not provided, they guarantee Lyapunov stability by constraining the choice of parameters of the controller. This was also the only article that appeared to address stability of the RL system, which is a significant real-world implementation challenge (Sect. 12).

Approximate dynamic programming (ADP) is a powerful modeling technique to solve practical problems where the central theme is stochastic optimization. ADP focuses on using the Bellman's equation, Powell (2009). Cui et al. (2022), Feng and Li (2022), Macek et al. (2017) have all successfully applied ADP for solving the industrial PdM problem.

Feng and Li (2022) applied ADP and RL to multistage production systems, to jointly optimize production and maintenance. The system cost is defined as the sum of inventory, backlog (production loss of the last machine) and maintenance costs. An optimal maintenance policy is one that minimizes the expected discounted cost over an infinite horizon. They use first-principles—a standard feed-forward neural-network function approximation for the value function, employing the Bellman's optimality equation, stochastic gradient descent and experience replay techniques. Comparison against three standard industry maintenance policies—state-based policy (SBP) i.e. maintenance based on a state determined by heuristics, time-based policy (TBP) and "greedy policy" i.e. run-to-failure based policy (RTFP) are performed using numerical simulations. RL demonstrates lower costs: 9.68%, 39.07%, and 39.56% over SBP, TBP and RTFP, respectively. Their methods additionally help identify bottlenecks achieving a 9% throughput improvement.

Similarly Huang et al. (2020) and Liu et al. (2020) too address interdependent machines in a serial production line with buffer levels. In Huang et al. (2020) the RL algorithm optimally suggested which particular machine must be serviced and reduced the average maintenance costs by 5%, 7% and 20% for SBP, TBP and RTFP respectively. While in Liu et al. (2020) the short-coming of conventional methods that usually consider only the first maintenance regime as a success factor is addressed. The proposed deep RL solution demonstrated a 30% higher maintenance success by proposing *follow-up* maintenance schedules as well.

RL has been applied to optimize availability of curing machines, for the rubber industry, by Senthil and Pandian (2022) and suggest when to perform short, medium and long-term maintenance. Availability is modeled as first-order differential equations in the form of state probabilities. Q-Learning is employed to iteratively compute the Q-values. RL is effective in providing higher overall equipment efficiencies (OEE): 95.19% short-term and 83.37% long-term.

10.3 Joint-optimization of multiple industrial problems

10.3.1 Integrated assembly and maintenance scheduling problem

Semiconductor manufacturing process is extremely *complex*. An excellent application of DQN for an integrated prediction of order dispatching and *opportunistic maintenance* scheduling is presented by Valet et al. (2022). An opportunistic maintenance strategy optimizes time of maintenance by minimizing the opportunity costs. A 5 group, 10 machine, assembly is simulated. Normal distribution is used to model time parameters such as time to equipment-related failures, time to repair them and time to preventive maintenance. A 9-level reward function is defined as demanded by the complexity of the process. State space is a vector of five groups of information—requesting machine, status of all machines, orders, mean buffer utilization, maintenance

information. Hoffmann et al. (2021), addressees maintenance resource scheduling using the PPO (Proximal Policy Optimization) algorithm. They model state-space using two "views"—machine and maintainer views. Each containing resource, maintenance and production related variables. Rewards are divided into "local", that considers the current situation and "global" that addresses the overall cycle time.

Yang et al. (2021, 2018) address joint optimization of scheduling jobs and PM tasks by modeling them as MDP processes.

10.3.2 Machine and human resource management problem

While most surveyed articles dealt with management of PdM purely as a machine resource problem; Ong et al. (2021a) and Ong et al. (2021b) address *human resource* constraints as well. The first article models the decision making and risk attitude of maintenance staff, while second uses PPO-LSTM to manage maintenance of equipment, the cost associated with maintenance and human resources by managing the decision making dynamics. They model the maintenance actions as "Hold", "Repair" and "Replace" and human emotions as "Calm", "Cautious" and "High Alert". They evaluated the model with real human participants and achieve 53% and 65% higher performance when compared to conventional PdM methods and noted a learning efficiency improvement of 73% when compared to Hoong Ong et al. (2020).

10.4 PHM—Prognostic Health Management of machines

PHM problems are generally addressed by researches by assigning a continuous variable health-indicator. This numerical index helps engineers make maintenance decisions.

Ong et al. (2021b), Lee et al. (2014), Chen et al. (2020) model the fault development using the simple exponential decay function (27).

$$F(t) = \mathrm{e}^{-\lambda t}. \tag{27}$$

In addition, Ong et al. (2021b) and Zhang et al. (2019) (see Sect. 9.3.3) use the health index model proposed in Saxena et al. (2008) via (22). Ong et al. (2021b) uses POMDP framework.

Knowles et al. (2011) represents the state of the component using its age and "health" conditions (based on raw or transformed sensor values). A positive reward ("profit") is returned if the component does not fail when a "no maintenance" action was suggested and if the component failed, a repair "cost" is deducted as a penalty (negative reward).

Alternatively, RL in PHM has been used to plan an optimal and feasible schedule so that all assets can be covered for preventive maintenance by Hu et al. (2021), Ling et al. (2018), Min and Chao (2012), Wang et al. (2015), Zhang and Tang (2022). Table 12 lists the application areas that researchers have addressed.

11 Algorithms applied by researchers (RQ-4) and data-sets used for evaluation (RQ-5)

Algorithmic and technical details of a number of articles were studied in the previous sections. Section 9.2 was devoted to the SMART family of algorithms designed specifically for solving the PdM problem. Detailed case-studies in Section 9.3 followed by Section 10 with an industry oriented study; covering rotating and milling machines, hydraulic actuators, joint optimization of industrial problems. Assembly lines and multi-machine systems were covered and Table 7 summarized implementation details across multiple articles. Finally, Table 8 summarizes select research articles by listing the PdM scenario they address, research challenges they tackle and the algorithmic techniques used to address these challenges.

The basic Q-Learning (including, DQN (Deep Q-network)), along with its algorithmic variants were the most widely used algorithms found in literature. This was followed by actor-critic based methods and policy-gradient methods, PPO (Proximal Policy Optimization) and DDPG, (see Fig. 18 and Table 13). MDP, followed by SMDP is the most applied modeling strategy (Table 15).

Hoong Ong et al. (2020) employ Prioritized Double Deep Q-Learning with Parameter Noise (Case-study—Sect. 9.3.1). Gosavi (2004a) created the asynchronous, policy-iteration based Q-P learning algorithm. This is especially useful for SMDP environments and has been shown to be effective by Min and Chao (2012) and Wang et al. (2014). SMDPs model stochastic control problems and Baykal-Gürsoy (2010) explain how they are ideal for optimization of reliability maintenance problems as shown in Hu and Yue (2003). SMDPs have been implemented by Adsule et al. (2020), Wang et al. (2014), Das et al. (1999) and the SMART family of PdM specific algorithms (Sect. 9.2).

Table 16 lists the data-sets researchers have employed to evaluate performance. NASA's "Turbofan Engine Degradation" data-set appears to be the most widely adopted, followed by the CWRU data-set.

12 Challenges and research gaps identified (RQ-6)

RL for PdM is a combination of two fields with some inherent challenges.

12.1 Inherent challenges of prognostics

"Paradox of prognostics" (Saxena et al. 2010a)—to get accurate data, one must let the equipment fail, despite predictive signals. On the other hand taking a corrective action eliminates the chance of validating the prediction. Prognostics, by nature is an *acausal* problem. To predict accurately one needs knowledge of future events, for instance the exact operational conditions. Secondly, to evaluate the predicted EoL (end of life) accurately, the actual EoL must be known, which is impossible. All this adds uncertainty to the overall predictive process.

12.2 Inherent challenges of implementing deep RL

Numerous studies have been conducted by Google, MIT and Berkeley related to stability of RL algorithms, in Song et al. (2019), Henderson et al. (2018), Hardt et al. (2015), Zhang et al. (2018). Dulac-Arnold et al. (2021) outline *implementation* challenges.

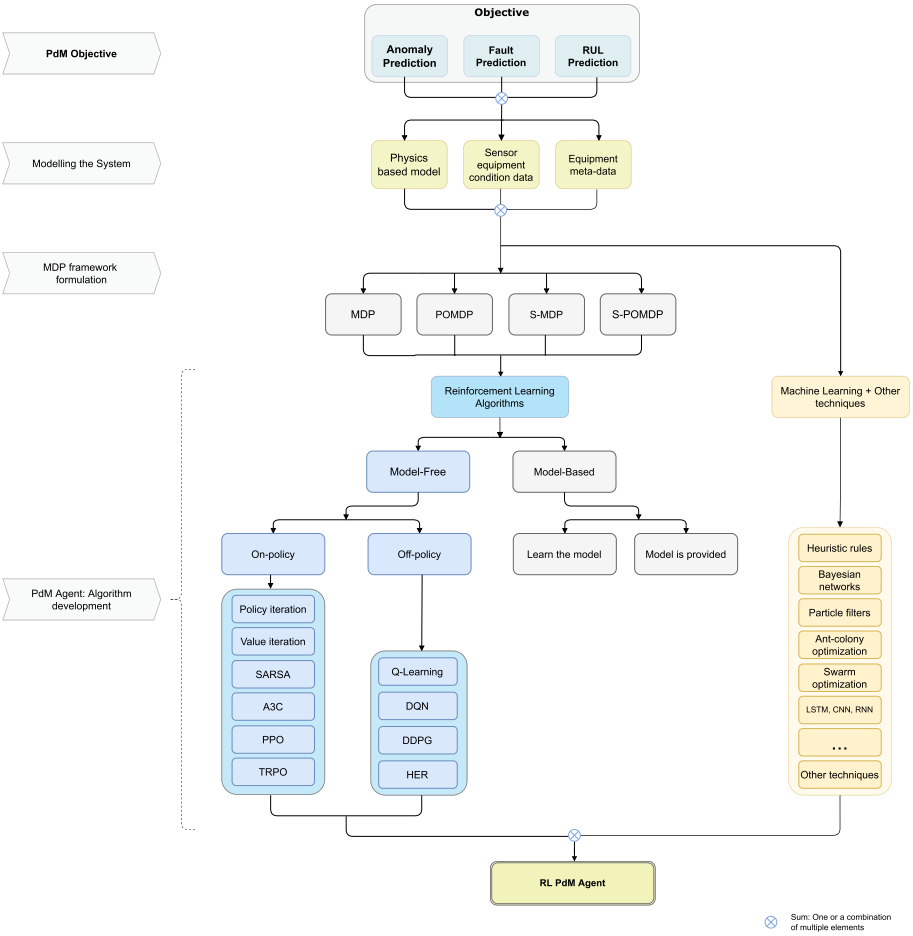


Fig. 14 A taxonomy of RL for predictive maintenance

Reward shaping and sensitivity to scale is a widely identified challenge in several articles (Fink et al. 2020; Grzes 2017). Sparse rewards are extremely common in PdM situations and it is exactly this and large reward values that have been attributed to network saturation by Duan et al. (2016). One can often stabilize performance by normalizing the reward range to `[0, 1]`.

Over-fitting in model-free RL is studied in Song et al. (2019). Their research indicate that, during exploration, often agents correlate rewards with spurious observation-space features. In a related study Hardt et al. (2015) provide theoretical proof, that stochastic gradient methods when trained using *fewer* iterations achieve more robustness and argue that simply reducing *training time* could prevent over-fitting.

Hyperparameter tuning significantly affects deep RL performance. Henderson et al. (2018) studied reproducibility of and sensitivity to network-architecture hyperparameters and rewards scaling, for model-free, PG algorithms. They conclude that while the simple ReLU activation performed best, their effects were inconsistent across algorithms and hyperparameter settings.

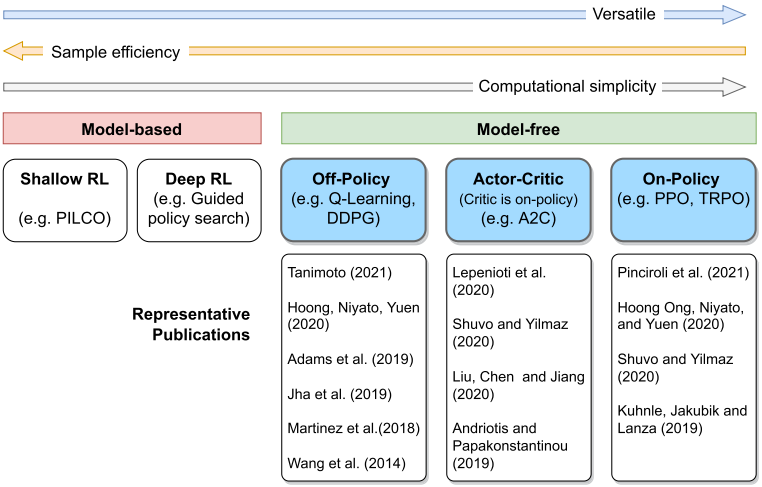


Fig. 15 A complexity/versatility view of RL algorithms. Enhanced from source: Hui (2021)

For predictive maintenance formulations where continuous action spaces are considered, a noise parameter is often added in an attempt to enable exploration and generalize learning. Plappert et al. (2017) observe that contradictory to general perception this is *not* necessary. On similar lines, sensitivity to random seeds was studied by Henderson et al. (2018) and they conclude that stochasticity in environment compounded with stochasticity of the learning process often lead to incoherent inferences (despite results averaged across multiple trials).

12.3 Selection of RL algorithm

Designing an optimal, model-free RL solution for PdM, has been shown to outperform human heuristic based and traditional ML solutions. Our survey shows an absence of guidelines that can assist the selection of RL algorithm, the type of MDP formulation, state and action representations and designing a reward function. This is even more important in the presence of RL challenges listed above. Together, the taxonomy, algorithm comparisons and suggested approach, presented in this survey, can all assist in the choice of direction to take—Figs. 14 and 15, Sect. 14 and Table 4.

Q-Learning (including DQN) accounted for over half (54.5%) of the studied articles. Only 3 researchers used PPO, while 2 used DDPG. Maintenance actions such as "perform maintenance" are discrete in nature for which Q-Learning is suitable. Policy gradient methods are suitable for *continuous* actions. Algorithms such as DDPG, PPO and TRPO (Trust Region Policy Optimization), could suggest continuous actions of the form "hours/days to next maintenance" or even "quantity of lubricating-oil".

12.4 State representation and uncertainties

Often system models have been simple and linear, for example (Jha et al. 2019b). For the agent to have a successful transfer of policy to a real environment, true, often highly

Table 4 Selecting an RL algorithm

Q-learning based	Policy-gradient	Actor-critic
Observation space		
Discrete or continuous	Discrete or continuous	Discrete or continuous
Action space		
Discrete	Discrete or continuous. (except DDPG which only support continuous)	Discrete or continuous
Advantages		
Simple implementation, sample efficiency and stability. Lower variance	Convergence guaranteed	Better sample efficiency than PG methods
Disadvantages		
Value-based approach Works for environments with *discrete* and *finite* state and action spaces. Require extensive hyperparameter search. Convergence not guaranteed	Policy-based approach. 10 times less sample efficient than Q-Learning. Exhibits higher variance therefore longer training time than A-C methods	Computationally intensive as there are *two* networks to train (Actor: *PG* method for π.. Critic: Approximates V)
Applied by		
Jha et al. (2019b), Hoong Ong et al. (2020), Martinez et al. (2018), Tanimoto (2021), Wang et al. (2014) and Adams et al. (2019)	Pinciroli et al. (2021), Hoong Ong et al. (2020) and Shuvo and Yilmaz (2020)	Shuvo and Yilmaz (2020), Andriotis and Papakonstantinou (2019) and Lepenioti et al. (2020)

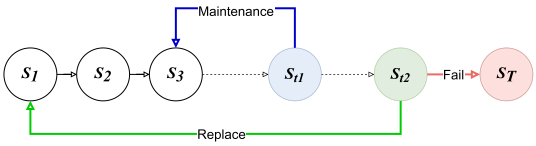


Fig. 16 State diagram with possible PdM states

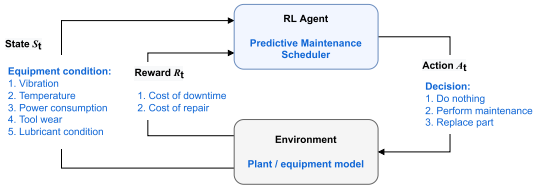


Fig. 17 The PdM problem formulated as an RL problem

non-linear, degradation processes must be modeled. To achieve this, we suggest the use of real world data to build the simulated environment by first learning the model of the environment (Sect. 14). The Stanford helicopter RL control study, Ng et al. (2006), used this technique to learn the stochastic, non-linear model using just 391 s of real flight data. Equipment degradation modeling is however more difficult. We need data from equipment that is degrading and allowed to run-to-failure. This could be practically difficult or expensive to obtain.

Real state representations must account for uncertainties due to environment stochasticity and measurement variations. Modeling the environment with "partially observable" states using POMDPs can tackle this as seen in Andriotis and Papakonstantinou (2021). Huang et al. (2019) highlights the main challenge, that of combining a machine's current status with its aging process to form a complete state representation.

12.5 Curse of dimensionality

State-action combinations can explode exponentially as systems become complex. Non-tabular methods such as DQN and function parametrization using neural-network methods is the most effective way to tackle this. 11% articles implemented multi-agents to help address decentralized control.

12.6 Moving from lab to field

None of the articles surveyed mentioned testing the agents in real world settings. This field has a practical difficulty—RL requires an environment to train and one cannot perform trial-and-error experiments on a real assembly line. Simulated environments are a standard practice. This creates a gap; the agent learns in a simulated environment but must work on a real factory floor. The incorporation of knowledge from field experts is vital. This can be in the form of heuristics and modeling parameters that represents real life equipment better. For example Wang et al. (2022) showed that their use of SMDP and *heuristically* accelerated multi-agent RL (HAMRL) is better than neighborhood search and "simulated annealing search" based exploration RL algorithms by 18.42% and 43.93% respectively. Zhang and Tang (2022) used multiple heuristics (15 constructive and 7 meta-heuristics) for an assembly line application, achieving reduction in total maintenance time by 45.16%.

Table 5 SMART evolution

Year	Algorithm and source	Description	Performance
1997	SMART (Mahadevan et al. 1997)	Semi-Markov (SMDP) based, model-free average-reward algorithm that seeks to maximize the net profit earned per unit time and suitable for *continuous*-time	SMART displayed higher average-reward than two heuristic policies COR (Coefficient of Operational Readiness) and AR (Age Replacement), it however resulted in *higher failures* trying to optimize maintenance cost (for higher maintenance costs)
1999	SMART (Das et al. 1999)	Convergence proof of SMART over an infinitely long time horizon	Mean values of average-reward was within 4% of the optimal value
2004	R-SMART (Gosavi 2004b)	Similar to SMART except relaxation scheme used for updating average reward ρ. Disadvantage: Exploration rate is required to undergo decay, failing which the algorithm may not converge and generate the optimal or near-optimal solution. The R-SMART is therefore limited by the tuning parameter that controls how the exploration rate is gradually reduced	Average reward ρ 0.0286 $\pm$ 0.0006
2017	iSMART (Encapera and Gosavi 2017)	Uses constant exploration rate unlike R-SMART and is more *robust* than it. iSMART produces a dual "image" of the main Q-factors (called R- and T-factors)	R-SMART ρ 0.0286 $\pm$ 0.0006 iSMART ρ 0.0373 $\pm$ 0.0002 i.e. $\uparrow$ 30.4% increase over R-SMART

Table 6 Bearing faults: early detection on CWRU data-set

Article	Technique	Accuracy[a]
Ding et al. (2019)	Base-line reference *(non RL)*: SAE-Softmax	0.9309
Ding et al. (2019)	Q-Learning with SAE	0.9249
Eltotongy et al. (2021)	CNN + RL + NAS	0.9934

[a]Average stage classification accuracy on Test set

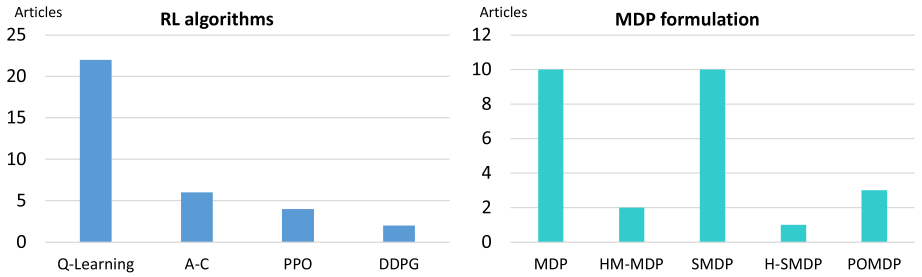


Fig. 18 Prominent environment modeling techniques and RL algorithms

Mathworks' Simulink and OpenAI's Gym environment are well known general RL environments. Nowadays, several commercial and open-source simulation platforms are available that are built for industrial settings, Table 9. These either have native RL agent integration capability or allow indirect integration via APIs. These may provide a more realistic environment, as seen in Fig. 19, for simulating environments or testing PdM policies and tune them further, before they are taken to field.

In Sect. 13, we suggest designing agents for the real-world by applying methods such as "**Meta-RL** and **robust reinforcement learning**".

12.7 Lack of consistency in performance evaluation

ML fields such as "classification" have matured metrics such as F1-scores, enabling comparison across research. On the other hand, for RL applied to PdM, both evaluation data and evaluation metrics varied extensively, making it difficult to compare superiority across research.

Some researchers used custom simulated data-sets, while some used public data-sets and yet others have compared against heuristic policies. The metrics used showed a wide range of variety with the average reward, being the most frequently adopted. This however depends on the reward function and scale used and therefore difficult to compare across researchers. Other articles compared performance using *statistical* metrics such as "action agreement %" (Andriotis and Papakonstantinou 2019), "relative deviation %" (Zhang and Tang 2022), classification accuracy (Ding et al. 2019; Eltotongy et al. 2021), geometric-mean (Dangut et al. 2022), average error rate (Chen et al. 2021) and some used *industrial* measures such as OEE (Senthil and Pandian 2022).

In Sect. 13.8, we provide suggestions on use of specialized metrics for PdM applications.

Table 7 Maintenance for assembly-lines and multi-machine systems: Comparison across implementations

Use-case	Source	Algorithm	Agent	MDP	Performance evaluation and notes
Serial production line	Huang et al. (2019)	Q-Learning/ DQN	Single-agent	MDP	Each machine has 4 degradation states and one failure state. 3-machine-2-buffer simulation. 70% improvement over CM
Serial production line	Huang et al. (2020)	Q-Learning/ DQN	Single-Agent	MDP	State: 3 elements (1) machine age (specified as probability of random failures on each machine); (2) buffer levels (3) remaining maintenance duration. 6-machine-5-buffer simulation. Reduced maintenance cost by 8.77% and 6.25% over age-dependent PM and OM respectively
Large-scale manufacturing systems with intermediate buffers	Li and Zhou (2020)	Q-Learning. Genetic Algorithm + multi-agent RL (GA+MARL). GA to guide the decision making of each agent	Multi-agent	MDP	Compared with MARL, average revenue rates ($\mu \pm \sigma$): GA = 0.7383 $\pm$ 0.0129; RL = 0.7374 $\pm$ 0.0152 and GA+RL = 0.8693 $\pm$ 0.0155
Industrial edge-based IoT	Ong et al. (2021b)	PPO with LSTM	Single-agent	MDP	Compared to conventional RL methods $\uparrow$ 53% and *game-based* comparison against human participants $\uparrow$ 65%. On NASA C-MAPSS shows learning efficiency improvement of $\uparrow$ 73%
Pumping system RUL	Bellani et al. (2019)	Q-Learning	Single-agent	MDP	Compared to two *heuristic* rules. Avg. rewards *R*: Low performance setting: $R = 40.93 \uparrow 2.44\%$; High performance setting: 41.13, and RL 41.93 $\uparrow$ 1.95%
Bearings. Identifying "change of health" points	Cheng et al. (2018)	Q-Learning and reward based on Calinski-Harabaz (Caliński and Harabasz 1974) clustering index	Single-agent	MDP	PRONOSTIA data-set (Nectoux et al. 2012). SVM and k-means incorrectly "clustered" vibration into 4 levels based while RL correctly identified the 4 *time*-axis stages
Aerospace long-term maintenance	Hu et al. (2021)	Q-Learning + Extreme Learning Machine	Single-agent	MDP	Avg. rewards RL 210.45, $\uparrow$ 189% compared with PM and $\uparrow$ 127% CM

Table 7 (continued)

Use-case	Source	Algorithm	Agent	MDP	Performance evaluation and notes
DC motor—shaft wear RUL	Jha et al. (2019b)	Q-Learning	Single-Agent	MDP	Modeling: Localized wearing using Archard equation (Meng and Ludema 1995). State:Current and shaft-speed; Action: 0 to 10 V in 20 levels. Learns offline control
Wear of component layer thickness	Adsule et al. (2020)	Q-Learning based SMART (Mahadevan et al. 1997)	Single-agent	Continuous time SMDP	Demonstrated with 100 states. Learns optimal policy in about 1000 steps

Table 8 Algorithm details: PdM scenarios and challenges addressed by researchers

Article	PdM scenario	Challenges addressed	RL algorithm and techniques applied
Andriotis and Papakonstantinou (2021)	Multi-component deteriorating system. 4 levels of state-damage with 50 deterioration rates	(1) Planning under incomplete information and constraints (2) Curse-of-dimensionality due to multi-component (3) Explosion in number of decision-trees (4) State uncertainties and environment stochasticity (5) Measurement gauge variability	Deep Decentralized Multi-agent Actor Critic (DDMAC)—consisting a combination of constrained Partially Observable MDP (POMDP) and multi-agent Deep RL. DDMAC addresses challenge (1) via function parametrizations and decentralized control. POMDPs combine stochastic dynamic programming and Bayesian inference to address (2) and (3). Proper state augmentation and Lagrangian relaxation address (4) and (5)
Afshari et al. (2014)	Estimate RUL of roller bearings, based on crack propagation	Non-linear Recursive Least Squares (RLS) method have a fixed step-size and are not suitable for estimation of non-stationary stochastic systems	(1) Paris's formula used for the defect propagation model. (2) An adjustable step size increases the estimation performance (3) Auto-step RL algorithm allows smoother parameter estimation along with a smaller variation in the estimated defect area
Skordilis and Moghaddass (2020)	Turbofan system. Estimate RUL for degrading systems. Derive maintenance policies in real-time	(1) Real-time control. (2) Address uncertainties in the system dynamics without making too many distributional and parametric assumptions. (3) Unavailability of fully observable systems	Algorithm based on combination of Deep RL and particle filtering. (1) State modeling using Hybrid State-Space Models (HSSM) (2) Latent states of HSSM are not observable over time and require the computation of posterior distributions. (3) Particle filters enable estimation of posterior density of state variables given the observation variables. (4) RL agent actions are chosen based on a stochastic environment with states inferred by the Bayesian filter

Table 8 (continued)

Article	PdM scenario	Challenges addressed	RL algorithm and techniques applied
Wang et al. (2014)	Single machine with multiple deteriorating yield levels	(1) Handling imperfect *minor* maintenance while assuming perfect major repairs. (2) Situations where the elementary yield levels cannot be directly obtained	Q-P algorithm (Gosavi 2004a). State modeled using product *quality inspection* information. Uses hidden semi-Markov (HSMDP). Agent learns a maintenance policy which is used to estimate future maintenance time by *re-simulating* the system
Yan et al. (2019)	Replacement of lubricating oil in machines used in oil-fields. Optimal time for replacing the lubricating oil and hence reduce unnecessary expensive downtime	(1) How to use *multiple* location oil spectral analyses to create a single health-index (HI) measure for lubricating oil condition. (2) Oil spectral analysis provides 15 types of main element concentrations, where different element concentrations have different physical significance	(1) Degradation of lubricating oil, contaminated with debris, is modeled as a continuous-time stochastic **Wiener process** (WP) (2) HI for the lubricating oil is based on **Shannon entropy** a concept from information theory. This assists in quantitatively selecting degradation data that contains most health related information from oil spectral analysis. (3) Weighted average of multiple point spectral data. Measure the *relative contribution* rate of each data-set to the overall lubricating oil degradation. Limitation: Linear weigthages used and nonlinear functions may provide better inference. (4) Based on evaluated oil HI oil replacement problem is modeled as MDP. (5) RL value iteration algorithm applied to determine optimal decisions

Table 8 (continued)

Article	PdM scenario	Challenges addressed	RL algorithm and techniques applied
Li and Zhou (2020)	Large-scale manufacturing systems with intermediate buffers. Series system with assembly and disassembly systems	(1) With increase in number of components, state and action space explodes. (2) As the number of agents increase, the complexity of reward function increases resulting in agents receiving *noisy* reward signals, causing difficulty in convergence to optimal strategies	(1) Multi-agent RL (MARL) for multi-component system. (2) Genetic algorithms (GA) to assist convergence via their global optimization abilities. (3) *Bi-directional interaction* mechanism between MARL and GA to enable sharing of best policies. (4) MARL exploration is based on ϵ-greedy technique
Cheng et al. (2018)	Degradation of bearing assemblies	*Simultaneously* address *multiple* health-indices (HI) and *change points*	(1) Apply RL to identify optimal health "change-points" (RLCP). (2) Feature engineering to extract six time-domain features as the multiple HIs: mean value (first moment), RMSE, crest factor, average power, skewness (third moment) and kurtosis (fourth moment). (3) Model the health stage division process as an MDP
Feng and Li (2022)	Multistage production systems, with multiple deterioration states and experiencing machine stoppage bottlenecks (MSB)	For multistage production systems, analytical identification of MSBs using standard definition (function of ratio of "unsatisfied market demand" (USMD) and downtime duration (DTD) of machines), is not feasible due to the absence of a closed-form expression	(1) RL for joint optimization of production and maintenance *cost*. (2) Integrates production system modeling and approximate dynamic programming (ADP). (3) USMD function is based on "starvation events", in turn based on rated speed and state of machine, to compute the MSBs. (4) "Bottleneck Improvement Method (BIM)"—iteratively reduce the downtime duration (DTD) of MSB by 5% until MSB transfers to next bottle-neck machine. (5) Standard RL Q-learning algorithm. Value function based on discounted cost

Table 8 (continued)

Article	PdM scenario	Challenges addressed	RL algorithm and techniques applied
Ling et al. (2018)	Two-machine flow-line with an intermediate buffer. System operating under maintenance resource constraints (financial, manpower and equipment)	(1) Independently deteriorating machines (2) Handling *imperfect* preventive maintenance, resulting in the machine being in an intermediate "health" condition	(1) Modified form of the model-free average-reward SMART (Das et al. 1999) algorithm, called "Distributed SMART" applied. (2) Degradation quality states of each machine represented by multiple independent decreasing yield levels. (3) Machine degradation processes formulated as a *continuous-time* SMDP to model each machine and its adjacent buffer, with *variable* time actions such as production and maintenance. (3) SMDP average reward algorithm implemented for *each* machine and its adjacent buffer. (4) Distributed SMART, is a *multi-agent* Q-Learning variant and solves the multiple decoupled sub-SMDPs, maintaining the relation between immediate *global* costs and the local decisions made by each agent, while adhering to the overall optimization goal

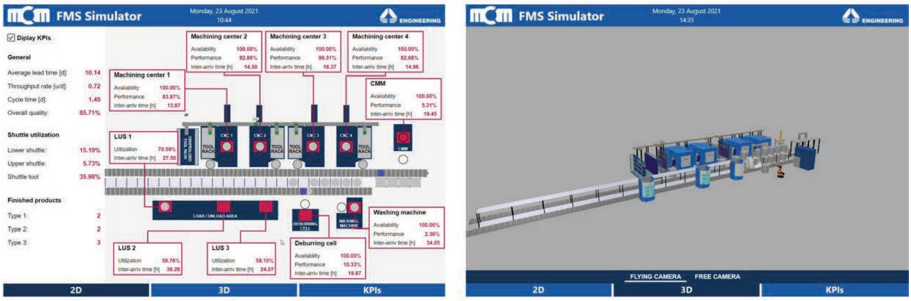


Fig. 19 AnyLogic™: commercial simulation software (reproduced with permission)

13 Opportunities and suggested future pathways (RQ-7)

13.1 RL methods for enabling agents to handle real-world environments

Moving an agent from a simulated environment to the real world can degrade the performance in the face of uncertainties, disturbances, or changes to the environment structure itself. This can lead to unpredictable results. In this section we highlight three RL methods that enable an agent to handle this movement – "**Robust RL** design", "**Risk aware RL**" and "**Meta-RL**". We did not find any article applying these methods to PdM and therefore these present an opportunity for future research.

Robustness of the agent is critically important and can be defined as its ability to cope with variations and uncertainties associated with its environment. Moos et al. (2022) provide an excellent technical foundation of **robust RL** and have conducted a review of existing literature, categorizing it into four design methods: (i) Transition design addresses uncertainties by manipulating the transition probabilities from one state to other; (ii) Disturbance design use external forces to model system uncertainty; (iii) Action design perturb the actions the agent performs; and finally (iv) Observation designs distort the state exploiting a vulnerable policy. Robust RL inspired by the $\mathcal{H}_\infty$ control theory has been covered by Morimoto and Doya (2005) and Pinto et al. (2017).

A related field is **risk-sensitive RL**. Humans are always weighing the decisions they take against the risks associated with future uncertainties in their environment. Shen et al. (2014) describe a risk-sensitive Q-learning algorithm that models human behavior. Fei et al. (2021) employ entropic risk measure while Prashanth et al. (2022) provides seven risk measures for a quantitative evaluation of risk and provide a risk based formulation of the objective.

Meta reinforcement learning (meta-RL) uses experience gained while performing previous tasks (as for example in the simulated environment) for solving a set of *new* tasks (or situations not encountered previously as may happen in the case of a real environment), Imagawa et al. (2022). Adapting prior knowledge addresses the problem of sample inefficiency. Hua et al. (2021) have designed a framework called Hyper Meta-RL (HMRL) to specifically address sparse rewards and is therefore applicable to the field of PdM.

Table 9 Simulation systems

Industry	Product/ link	Type	RL support	Notes
Manufacturing, oil-and-gas, energy, mining, logistics	AnyLogic	Commercial	Integrated, easy	Allows RESTful API based integration with custom Python code. Uncertainty and risk modeling
Production planning and control of complex assembly lines systems	SimRLFab	Open-source	Integrated, easy	Native Python low-level development
Robotics and other physics based simulations. Safety simulation	Gazebo	Open-source	Indirect, easy	Cloud based services. Unique *sensors* library (e.g. monocular cameras, depth cameras, LIDAR[a], IMU[b] etc.)
Chemical process simulator	DWSim	Open-source	Indirect, difficult	CAPE-OPEN: Computer Aided Process Engineering compliant
Components such as pumps and wind-turbines to systems such as buildings	TRNSYS	Commercial	Indirect, difficult	C, C++ integration

[a]Light Detection and Ranging

[b]Inertial Measurement Unit

13.2 Batch or offline RL

Running experiments or training in a live industrial setting will generally not be encouraged. Instead, collecting and using data offline is acceptable. **Batch Reinforcement Learning** (also known as **Offline Reinforcement Learning**) is an RL method where the agent is tasked to learn a policy from a fixed batch of data. In Batch RL, the data is provided a priori and there is *no exploration*. Model stability and data efficiency are the main advantages of this method. This method enables a more stable agent to move from a simulation environment to the real environment. We did not find applications of Batch RL to PdM while this seems to be a promising possibility. Lange et al. (2012), Nair et al. (2020) describe this method.

13.3 Complex systems made up of multiple sub-systems

We did not find a single article that addressed a machine as a system made up of heterogeneous sub-systems such as bearings, gears, shafts, belts, motor winding etc. Andriotis and Papakonstantinou (2019) does provide an excellent treatment of complex civil structures composed of sub-systems, which are albeit *homogeneous* (steel truss members). There exists an opportunity to attempt multi-agent RL, such that individual sub-systems have their own degradation models and maintenance requirements and aggregated at a system level. Batch RL, described earlier, is suitable for learning in multi-agent systems, Lange et al. (2012).

13.4 Reward shaping and reward learning through Inverse RL

Reward shaping is an active area of research. RL requires time to learn stable policies, since the agent must determine the consequences of their actions in the long term. "Reward shaping" (Grzes 2017) incorporates domain knowledge so that the algorithms are guided more efficiently. It is applicable to both model-free and model-based algorithms as well as multi-agents.

Combining this with another very active area of research, Inverse Reinforcement Learning (IRL), is potentially a very fertile ground for research. Inverse RL is essentially *learning the reward function*. We did not find any article that combined these two possibilities.

13.5 Curriculum learning for PdM

If as suggested above, one models complex systems as a collection of sub-systems or by using real data, RL will require significant efforts to tune and time to learn.

To address this challenge, a form of *transfer learning* known as Curriculum Learning (CL) (Narvekar et al. 2020) can be applied. The experience gained while learning one task can be transferred to the agent before it starts learning the next, harder task.

None of the articles surveyed applied Curriculum Learning to the PdM task. We suggest this as an area for further research. Researchers could look at not only structuring tasks, but also the training data (experiences) in a similar fashion (simple to hard).

13.6 Hierarchical RL and the options framework

Hierarchical reinforcement learning (HRL) enhances the performance of an agent on a complex task by using a "divide-and-conquer" approach. A difficult problem is divided into multiple sub-problems that are easier to solve. HRL uses two key mechanisms to achieve this—temporal- and state-abstractions. Systematized application of abstractions allow sample-efficient algorithms. An RL "Options framework" enables the discovery of these abstractions (Hutsebaut-Buysse et al. 2022). In Sutton et al. (1999), temporal abstraction are formalized by SMPDs. Options are closed-loop sub-policies (i.e. sub-behaviors) (Hutsebaut-Buysse et al. 2022; Sutton et al. 1999). This is another promising RL method we did not find applied to the PdM problem. These methods may help find temporal or state abstractions during the life of an equipment and could assist in predicting a possible maintenance action.

13.7 Maintenance of the controller

Controllers, such as the ubiquitous proportional-integral-derivative (PID) and model predictive control (MPC), are an integral component of industrial equipment and processes. While the focus of this survey has mainly been the predictive maintenance of industrial equipment, the application of RL for the predictive maintenance of *controllers* themselves have not been adequately studied.[8] Pro-actively identifying the drift in controllers, and initiating controller tuning could offer substantial reduction in reactive maintenance costs. Once the drift in control variable or process variable is predicted, via trend charts or analytical means, RL could be used to tune the PID controller. Controller parameters are dependent on the operational conditions. Dogru et al. (2022) formulate PID tuning as an RL task. They have applied *off-line* agent training to identify an initial approximate step-response model and then subsequently fine-tune the PID *on-line* on the actual process thereby adapt and re-tuning to the real process dynamics. Kofinas and Dounis (2019), Shi et al. (2020) provide additional examples of how RL can be used for autonomous tuning of PID, while Mehndiratta et al. (2018) applies RL to automatically tune a nonlinear MPC.

13.8 Evaluating performance of PHM applications

One of the challenges uncovered was the lack of consistency in evaluating the performance of PdM solutions. The metrics used were statistical measures suitable for general predictive models and not specialized for prognostics. Prediction of RUL and EoL always involves uncertainty and the "paradox of prognostics" makes it further challenging. RUL is not a stationary value. Estimated RUL is true only for conditions prevailing at that time. Its estimate can be continuously updated as more data becomes available. Prognostics prediction deals with equipment failure and therefore becomes more critical as equipment approaches its EoL.

Saxena et al. (2010a, b) have presented an excellent treatment on evaluating PHM applications and is suitable for PdM based on RUL or a health-index. They suggest two

[8] As of 09-Feb-2023, the search "`'reinforcement learning' AND 'predictive maintenance' AND (PID OR MPC)`", did not return any results on the Scopus and Web Of Science databases.

important concepts to represent uncertainty and distance from true EoL using α and λ. α is the error-bound, specified in percent; α is use-case dependent, and can be based on the time required to perform corrective action. λ is a relative-to-EoL time measure (28), therefore $\lambda = 0.5$ specifies halfway to failure from first prediction.

$$t_\lambda = t_P + \lambda \cdot (t_{EoL} - t_P), \tag{28}$$

where, t_P the time of prediction and t_{EoL} is actual end-of-life

The following hierarchical flow of metrics is suggested (Saxena et al. 2010b, a):

1. **Prognostic Horizon**: Measures the time-index t_i when the agent is able make its *first* prediction, within the specified error bounds α.
2. α–λ **Accuracy**: Binary (True/False) metric. Given a point in time t_λ, measure if the performance can remain within specified error bounds α.

$$(1 - \alpha) \cdot r_*(t) \;\leq\; r^l(t_\lambda) \;\leq\; (1 + \alpha) \cdot r_*(t), \tag{29}$$

   where, l is the index under evaluation, r_* is the true RUL.
3. **Relative Accuracy**: Given a point in time relative to RUL, measure agent accuracy.
4. **Convergence**: If the above metrics are met, measure how soon this is achieved.

None of the surveyed articles applied these metrics, despite being valuable measures of performance. We encourage researchers to apply them.

13.9 State signal engineering with FFT, MFCC

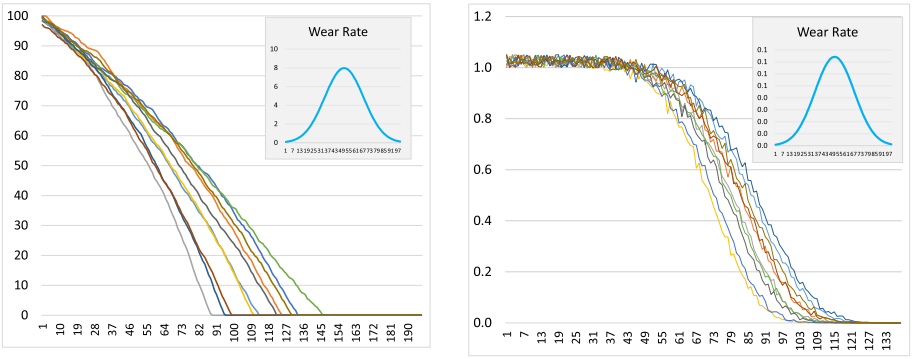
Signal processing techniques such as FFT (Fast Fourier Transforms) and MFCC (Mel-frequency cepstral coefficients) carry rich information especially for vibration signals. These can be used to augment raw sensor data for state signals in building the observation space. None of the surveyed articles applied these techniques and provide an opportunity for future research.

13.10 GAN and other techniques for augmentation of training experiences

RUL data is "expensive" and rare. Augmenting time-series data could help generate additional data representing similar distribution by application of GANs (generative adversarial networks) and adaptive weighting technique outlined in Fons et al. (2021). This technique could also save available real-data for training, while augmented data could be used for evaluation. We did not find any of the surveyed articles employing these techniques.

13.11 Stability of algorithm

As noted above, RL is known for being unstable suggesting that it is necessary to ensure stability of algorithms. Only Liu et al. (2017) addressed stability by applying constraints using Lyapunov stability criteria (applicable to non-linear dynamic systems (Alimi et al. 2021). Applying known and developing new methods for studying stability of algorithms is a suggested area of research.



(a) Linear degradation (b) Non-linear degradation

Fig. 20 Statistical modeling of component deterioration based on real data

13.12 Coverage of certain industries

Certain areas such as wind-turbines and aircraft seemed to be well researched. Industrial applications such as oil-and-gas, mining, waste-water treatment, industrial robots seem to be missing and present an opportunity for research. As are specific machines such as milling-machines, representing an area for focused development.

14 Conclusion

14.1 Practitioners guide to implementation

Based on our research, we now suggest a method of implementing RL for PdM, targeted toward industrial practitioners. Our objective is to provide a simple and *practical* approach and lay out guidelines that can be continuously built and improved upon in phases. We divide the process into four main steps: (1) basics, (2) designing the environment, (3) designing the state, action and rewards and finally (4) the training.

We use tool-surface wear as an illustrative example.

Step 1: Basics—Goal, RL algorithms and health-index definition:

1. **Goal**: Learn a policy that suggests a component replacement or maintenance, at an optimal time.
2. RL type: **Model-free**.
3. RL algorithm: **PPO** (Proximal Policy Optimization) with Clipped Objective for faster optimization.
4. Health-index: H as a function of time $H(t)$.

Step 2: Building a model of the environment
Using a real environment, where parts are allowed to fail is impractical due cost and actual time it may take to generate a single episode for training; simulation provides a

mechanism to train the model. To design a learning environment as close as possible to the real scenario the following method is suggested.

1. Collect *real* data related to degradation until failure (for e.g. flank wear of milling tools).
2. Visualize and model this data. Figure 20a shows wear following an approximately *linear* trend—for e.g. as in Adsule et al. (2020).
3. Fitting a linear statistical model[9] on the degradation data gives us a wear rate of say $w = 0.06$ mm/day.
4. For a component with a measured wear rate w; Fig. 20a shows how 10 simulations were created from a normal distribution created with a mean $w = 0.06$ and σ =0.05 mm/day (inset in Fig. 20a).
5. Taking this one step further, one can similarly model and then simulate non-linear degradation, Fig. 20b.
6. Use the degradation model to build an "environment".
7. In a model-free RL implementation, transition probabilities from one state to another are not known and will be learnt by trial-and-error via environment interaction.

Step 3: States, actions and rewards
Actions:

1. None (no maintenance interruption)
2. Maintenance (such as cleaning and lubrication)
3. Pro-active part replacement (preventive maintenance, PM)
4. Reactive part replacement (corrective maintenance, CM)

State:

1. Start with a direct health state indicator H e.g. tool surface wear measured in mm every hour or end-of-shift (represented by unit time t).
2. Example: A new tool piece has surface thickness of 5.0 mm and a replacement is required when the thickness wears down to below 1 mm. H can simply be the thickness normalized[10] to a scale between 0.0 (failed state) and 1.0 (best-health state) and updated using a simple linear function $H(t) = H(t-1) - w \cdot t$.
3. Improvement: Add features such as vibration, surface temperature, current drawn etc.

Rewards:

1. Start with a simple reward function, that penalizes for a lower health index
2. $\gamma = 0.95$ is suggested for visibility near equipment EoL

[9] This statistical model is different from the MDP "model" we refer to in the model-based/model-free RL.
[10] This assists in stabilizing learning in neural-networks.

3. $R = -1/(H(t) + \lambda)$. λ is a small constant (e.g. 10^{-3}) to avoid divide-by-zero when $H = 0$.
4. With a normalized $H \in [0, 1] \Rightarrow R \in [-1000.00, -0.99]$.
5. Improvement: Add maintenance cost C_{PM} to ensure maintenance events are optimized.
   $R = -C_{PM}/(H(t) + \lambda)$. It is assumed that cost of CM $\gg$ cost of PM.

Training: An episode ends on tool failure and its replacement. At least 2–4 thousand training episodes will be required to achieve a stable policy.

14.2 Research insights

This technical survey studied more than 100 articles. A summary of insights is presented in Table 10.

14.3 Concluding thoughts

In the era of Industry 4.0 and beyond, predictive maintenance is a significant lever in realizing a true SmartFactory. Evolution of PdM techniques along with a taxonomy of techniques was created. We presented technical details of some papers to demonstrate how they applied RL to solve the PdM problem.

One of the shortcoming identified by our survey was the absence of guidelines that can assist implementation engineers. Towards this, we presented a suggested approach and a detailed taxonomy that can assist and provide direction.

Some algorithms like SMART, developed specifically for PdM were described in detail. Analysis of algorithms show that Q-Learning (discrete action space) is the most widely used algorithm and the MDP process formulation. Use of simulated data remained predominant, that can make moving from lab to the field challenging. Challenges identified across the papers included lack of consistency in evaluation metrics and we suggest the use of Prognostic Horizon and α–λ Accuracy.

For further research we suggest modeling an equipment or machine system as a combination of sub-systems and reward shaping using heuristics and domain knowledge, use of Inverse RL and Curriculum Learning. All this should enable moving from lab to field more realistic. This can further be enhanced by augmentation of learning experiences. A significant suggestion is the use of "continuous-actions" algorithm such as DDPG, to suggest predictive maintenance actions such as quantity of lubrication oil to be applied or "hours until next maintenance".

We believe the important field of predictive maintenance will benefit from the autonomous learning mechanisms of RL and continue to evolve.

Table 10 Insights and suggestions

Survey findings and statistics		
1	Most prevalent industrial application covered	Production and assembly line systems 26%, followed by rotating machinery (includes bearing housings, aerospace and wind-turbines) 26%
2	Most prevalent model formulations	MDP 40% and SMDP 40%; covering 80% jointly
3	Most prevalent algorithm	Q-Learning and variants DQN, Double DQN
4	Most prevalent *public* data-sets	4 of 10 used NASA's Turbofan Engine Degradation data-set
5	Most prevalent evaluation metric	None. No consistency in metrics used
Insights and suggestions		
1	Selection of RL algorithm	Taxonomy and algorithm selection assistance: Figs. 14 and 15 and Table 4
2	Practitioners guide to implementation	Sect. 14. Use real data for environment
3	Simulation mechanisms	Table 9
4	RL for PdM: Performance evaluation	PdM/PHM specific metrics suggested: α (error-bound %)—time required for corrective action; λ—relative-to-EoL time measure; Prognostic Horizon—time when the first prediction is made (within α error-bound) and $\alpha-\lambda$ accuracy
5	State representation under presence of uncertainties	1. Consider applying "Curriculum Learning" by breaking down the PdM problem
		2. Consider modeling complex systems as multiple sub-systems and use multi-agent techniques
		3. Consider GAN and other techniques for augmentation of training experiences
6	Designing effective reward functions	1. Consider incorporating domain knowledge
		2. Consider *learning* the reward function using Inverse RL
7	Stability of algorithm	Consider applying "Curriculum Learning"
8	Missing coverage of certain industries	Oil and gas, mining, waster-water treatment and industrial robots

Appendix 1: Acronyms and notations

See Table 11.

Table 11 Acronyms and notations

Maintenance related terms			
CBM	Condition Based Maintenance	CM	Corrective Maintenance
EoL	End-of-life	KPI	Key Performance Indicator
MIMO	Multiple-input multiple-output	PdM	Predictive Maintenance
PdM	Predictive Maintenance	PHM	Prognostic Health Management
PHM	Prognostic Health Management	PM	Preventive Maintenance
RTFP	Run-to-failure based maintenance policy	RUL	Remaining useful life
SBP	State-based maintenance policy	TBP	Time-based maintenance policy
Algorithms			
CNN	Convoluted Neural Network	LSTM	Long Short Term Memory
A-C	Actor-Critic	PG	Policy Gradient methods
DDPG	Deep Deterministic Policy Gradient	PPO	Proximal Policy Optimization
DQN	Deep Q-Network	DDQN	Double Deep Q-Network
Reinforcement Learning			
ML	Machine Learning	RL	Reinforcement learning
MDP	Markov decision process	HM-MDP	Hidden-mode MDP
POMDP	Partially observable MDP	SMDP	Semi-MDP
Notations			
S	Set of all valid states	A	Set of all valid actions
s_t	State	s', $s_{(t+1)}$	Next state
a_t	Action	a', $a_{(t+1)}$	Next action
P, $P(s' \mid s, a)$	Probability distribution of state transitions	x	x
$R(t)$	Reward function	γ	Discount factor
π	Policy	π_θ	Policy π with learnable parameters θ
$V^\pi(s)$	Value function	$Q^\pi(s, a)$	Action-value function (Q value)
C_{CM}, c_i^{CM}	Cost of CM	C_{PM}, c_i^{PM}	Cost of PM
$H(t)$	Health index as a function of time t	$R(t)$	Weibull model: Probability of survival

Appendix 2: Tables—applications, algorithms and evaluation data-sets

See Tables 12, 13, 14, 15 and 16.

Table 12 Industrial applications (RQ-3)

	Industry/application	Articles
1	Aerospace	Ahmed et al. (2018), Dangut et al. (2022), Hu et al. (2021), Li et al. (2019) and Skordilis and Moghaddass (2020)
2	Battery systems	Chen and Li (2011) and Wu et al. (2021)
3	Civil engineering (bridges, large systems)	Andriotis and Papakonstantinou (2019), Cheng and Frangopol (2021), Shi et al. (2019) and Yang (2022)
4	Chemical industry	Wang et al. (2021b)
5	Coal industry	Liu et al. (2020)
6	Edge-based sensor networks	Hoong Ong et al. (2020)
7	Gas turbines (gas path faults)	Compare et al. (2020) and Luo (2021)
8	Hydraulic actuators	Adams et al. (2019)
9	Infrastructure maintenance (complex and large-scale)	Tanimoto (2021)
10	Iron and steel factories	Lepenioti et al. (2020)
11	Military (trucks—fleet maintenance)	Barde et al. (2019)
12	Milling machines (cutter tool wear)	Dai et al. (2021)
13	Lubrication oil	Yan et al. (2019)
14	Oil and gas	Compare et al. (2020)
15	Petroleum industry	Aissani et al. (2009) and Min and Chao (2012)
16	Planning of maintenance inventory	Das et al. (1999) and Xanthopoulos et al. (2017)
17	Power systems and grids (industrial)	Rocchetta et al. (2019) and Shuvo and Yilmaz (2020)
18	Production and assembly lines (multiple machine and MIMO systems)	Cui et al. (2021, 2022), Huang et al. (2019, 2020), Ling et al. (2018), Liu et al. (2017), Min and Chao (2012), Paraschos et al. (2020), Senthil and Pandian (2022), Su et al. (2022), Wang et al. (2014, 2015, 2016), Zhang and Tang (2022), Zhang and Si (2020) and Zheng et al. (2017c)
19	Rotating machinery (bearing housings)	Afshari et al. (2014), Dai et al. (2020), Ding et al. (2019), Dong et al. (2021a) and Skordilis and Moghaddass (2020)
20	Robotics	Li et al. (2021)
21	Rubber industry (curing machines)	Senthil and Pandian (2022)
22	Semiconductor manufacturing	Hoffmann et al. (2021), Valet et al. (2022) and Scheibelhofer et al. (2012)
23	Solar systems	Correa-Jullian et al. (2020)
24	Wind turbines and wind farms	Dong et al. (2021b), Eltotongy et al. (2021) and Pinciroli et al. (2020, 2021, 2022)

Table 13 Algorithms applied by researchers (standard and their variants)

Algorithms	Articles
1. Q-Learning (including Deep Q-network (DQN))	Adams et al. (2019), Cheng and Frangopol (2021), Correa-Jullian et al. (2020), Dai et al. (2020), Ding et al. (2019), Epureanu et al. (2020), Dong et al. (2021b), Hoong Ong et al. (2020), Hu et al. (2021), Huang et al. (2019, 2020), Jha et al. (2019a, 2019b), Martinez et al. (2018), Rocchetta et al. (2019), Senthil and Pandian (2022), Skordilis and Moghaddass (2020), Tanimoto (2021), Wang et al. (2021a), Zhang and Tang (2022), Zhang et al. (2021) and Zhang and Si (2020)
2. Actor-Critic based algorithms	Andriotis and Papakonstantinou (2019), Hosseinloo and Dahleh (2021), Liu et al. (2017, 2020), Shuvo and Yilmaz (2020), Su et al. (2022) and Wang et al. (2021b)
3. PPO	Kuhnle et al. (2019), Ong et al. (2021b) and Pinciroli et al. (2021, 2022)
4. DDPG	Chen et al. (2022) and Hosseinloo and Dahleh (2021)
Complex formulations	
5. Multi-agent formulation	Andriotis and Papakonstantinou (2019, 2021), Kuhnle et al. (2019), Su et al. (2022) and Wang et al. (2016)
6. Multi-objective formulation	Lepenioti et al. (2020)

Table 14 Specialized and novel algorithms applied by researchers

Algorithms	Articles
1. Q-P learning algorithm designed by	Gosavi (2004a)
2. Q-P learning algorithm applied by	Min and Chao (2012) and Wang et al. (2014)
3. Health Indicator Temporal-Difference Learning with Function Approximation (HITDFA)	Zhang et al. (2019)
4. RL for Health Stage Change Points (RLCP)	Cheng et al. (2018)
5. RL and particle filtering	Skordilis and Moghaddass (2020)
6. RL for neural architecture parameter search (NAS)	Eltotongy et al. (2021)
7. RL and continuous wavelet transforms (CWT)	Eltotongy et al. (2021)
8. Reinforcement Lion Swarm Optimization Algorithm	Dai et al. (2021)
9. RL and Petri Nets	Mao et al. (2021)
10. Markov Mixed Membership Models (MMMM) and POMDP	Hofmann and Tashman (2020)

Table 15 MDP model formulation

Markov formulation	Articles
1. MDP	Andriotis and Papakonstantinou (2019), Barde et al. (2019), Chen et al. (2022), Gosavi (2004b), Gosavi and Parulekar (2016), Mikhail et al. (2019), Shi et al. (2019), Yang et al. (2018, 2021) and Paraschos et al. (2020)
2. HM-MDP (Hidden Mode MDP)	Yang and Qi (2013) and Hofmann and Tashman (2020)
3. SMDP (Semi MDP)	Gosavi (2004b), Wang et al. (2014), Das et al. (1999), Encapera and Gosavi (2017), Hosseinloo and Dahleh (2021), Wang et al. (2015, 2016, 2021c), Ling et al. (2018) and Adsule et al. (2020)
4. H-SMDP (Hidden Semi MDP)	Wang et al. (2014)
5. POMDP (Partially Observable MDP)	Andriotis and Papakonstantinou (2019, 2021) and Hofmann and Tashman (2020)
6. Markov Mixed Membership Models (MMMM) and POMDP	Hofmann and Tashman (2020)

Table 16 Environments and data-set used for evaluation

Evaluation data-set	Publications
1. Turbofan Engine Degradation Simulation Data Set, NASA (Saxena and Goebel 2008). Engine run-to-failure time-series *simulated* data (using C-MAPSS tool). Four different data-sets simulated under different combinations of operational conditions and fault modes	Skordilis and Moghaddass (2020), Ong et al. (2021b), Hoong Ong et al. (2020) and Zhang et al. (2019)
2. CWRU (Case Western Reserve University) bearing failure data-set (Li 2019)	Ding et al. (2019) and Eltotongy et al. (2021)
3. PRONOSTIA bearing failure data-set (Nectoux et al. 2012)	Cheng et al. (2018)
4. UCR Timeseries (Dau et al. 2019). Three data sets are used for evaluation: (1) Gun-point (2) Wafer (3) ECG. Of these only the "wafer" data-set is related to the PdM field and contains inline process control measurements, classified as normal and abnormal	Martinez et al. (2018)
5. Milling cutter run-to-failure data for 3 cutters: force, vibration and acoustic emission. Wear measurement by LEICA MZ12 microscopy system (Prognostics and Society 2010)	Dai et al. (2021)
6. *Simulated* wind-farm of 50 wind-turbines using failure rate of wind-turbines from (Ozturk et al. 2018)	Pinciroli et al. (2021)

Appendix 3: *tf* − *idf* weighting

The $tf - idf$ scheme assigns a weight to each term in a document using Eq. (30), that is *low* when the term is infrequent in a document or occurs in many documents and is *high* when the term occurs many times within a small sub-set of documents.

$$tf - idf(t, d) = tf(t, d) \times idf(t),, \tag{30}$$

where t is a term within the document-set d; and the inverse document frequency, idf, is computed using (31)

$$idf(t) = log\left[\frac{n}{df(t)}\right] + 1 \tag{31}$$

Acknowledgements We would like to sincerely thank the Reviewers. Their valuable comments and suggestions helped us improve the technical quality of the manuscript.

Funding This work was supported by Symbiosis Institute of Technology.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Abernethy RB (2018) Dr. E. H. Wallodi Weibull. http://km.fgg.uni-lj.si/PREDMETI/sei/Ljudje/weibull.htm

Abudali M, Siegel D (2021) A pressing case for predictive analytics at Maclean–Fogg. https://www.plantengineering.com/articles/a-pressing-case-for-predictive-analytics-at-maclean-fogg/

Achiam J (2018a) Deep deterministic policy gradient—the q-learning side of DDPG. https://spinningup.openai.com/en/latest/algorithms/ddpg.html#the-q-learning-side-of-ddpg

Achiam J (2018b) Part 1: key concepts in RL—spinning up documentation. OpenAI. https://spinningup.openai.com/en/latest/spinningup/rl_intro.html#key-concepts-and-terminology

Adams S, Meekins R, Beling P et al (2019) Hierarchical fault classification for resource constrained systems. Mech Syst Signal Process. https://doi.org/10.1016/j.ymssp.2019.106266

Adsule A, Kulkarni M, Tewari A (2020) Reinforcement learning for optimal policy learning in condition-based maintenance. IET Collabor Intell Manuf 2(4):182–188. https://doi.org/10.1049/IET-CIM.2020.0022

Afshari H, Al-Ani D, Habibi S (2014) Fault prognosis of roller bearings using the adaptive auto-step reinforcement learning technique. In: ASME 2014 dynamic systems and control conference (DSCC 20140), p 1. https://doi.org/10.1115/dscc2014-5928

Ahmed I, Khorasgani H, Biswas G (2018) Comparison of model predictive and reinforcement learning methods for fault tolerant control. IFAC-Papers OnLine 51(24):233–240

Aissani N, Beldjilali B, Trentesaux D (2009) Dynamic scheduling of maintenance tasks in the petroleum industry: a reinforcement approach. Eng Appl Artif Intell 22(7):1089–1103. https://doi.org/10.1016/j.engappai.2009.01.014

Alimi M, Rhif A, Rebai A et al (2021) Optimal adaptive backstepping control for chaos synchronization of nonlinear dynamical systems. Backstepping Control of Nonlinear Dynamical Systems pp 291–345

Andriotis C, Papakonstantinou K (2019) Managing engineering systems with large state and action spaces through deep reinforcement learning. Reliab Eng Syst Saf 191:106483. https://doi.org/10.1016/j.ress.2019.04.036

Andriotis C, Papakonstantinou K (2021) Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. Reliab Eng Syst Saf 212(107):551

Bala R, Govinda R, Murthy CS (2018) Reliability analysis and failure rate evaluation of load haul dump machines using weibull distribution analysis. Math Model 5(2):116–122. https://doi.org/10.18280/mmep.050209

Barde S, Yacout S, Shin H (2019) Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. J Intell Manuf 30(1):147–161. https://doi.org/10.1007/s10845-016-1237-7

Barja-Martinez S, Aragüés-Peñalba M, Munné-Collado Í et al (2021) Artificial intelligence techniques for enabling big data services in distribution networks: a review. Renew Sustain Energy Rev 150(111):459. https://doi.org/10.1016/j.rser.2021.111459

Baykal-Gürsoy M (2010) Semi-markov decision processes. In: Wiley encyclopedia of operations research and management science. Wiley, Hoboken

Bellani L, Compare M, Baraldi P et al (2019) Towards developing a novel framework for practical PHM: a sequential decision problem solved by reinforcement learning and artificial neural networks. Int J Progn Health Manag 10(4). https://doi.org/10.36001/ijphm.2019.v10i4.2616

Ben-Daya M, Duffuaa SO, Raouf A (2012) Maintenance, modeling and optimization. Springer, Berlin

Burke R, Mussomeli A, Laaper S et al (2017) The smart factory. Deloitte Insights. https://www2.deloitte.com/us/en/insights/focus/industry-4-0/smart-factory-connected-manufacturing.html
Busoniu L, Babuska R, De Schutter B et al (2017) Reinforcement learning and dynamic programming using function approximators. CRC Press, Boca Raton
Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat Theory Methods 3(1):1–27
Chen H, Li X (2011) Distributed active learning with application to battery health management. In: 14th International conference on information fusion 2011
Chen Z, Wu M, Zhao R et al (2020) Machine remaining useful life prediction via an attention-based deep learning approach. IEEE Trans Ind Electron 68(3):2521–2531
Chen G, Liu M, Kong Z (2021) Temporal-logic-based semantic fault diagnosis with time-series data from industrial Internet of Things. IEEE Trans Ind Electron 68(5):4393–4403. https://doi.org/10.1109/TIE.2020.2984976
Chen Y, Liu Y, Xiahou T (2022) A deep reinforcement learning approach to dynamic loading strategy of repairable multistate systems. IEEE Trans Reliab 71(1):484–499. https://doi.org/10.1109/TR.2020.3044596
Cheng M, Frangopol D (2021) A decision-making framework for load rating planning of aging bridges using deep reinforcement learning. J Comput Civ Eng. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000991
Cheng Y, Peng J, Gu X et al (2018) RLCP: a reinforcement learning method for health stage division using change points. In: 2018 IEEE international conference on prognostics and health management (ICPHM 2018). https://doi.org/10.1109/ICPHM.2018.8448499
Coleman C, Damodaran S, Deuel E (2017) Predictive maintenance and the smart factory. https://www2.deloitte.com/content/dam/Deloitte/us/Documents/process-and-operations/us-cons-predictive-maintenance.pdf
Compare M, Bellani L, Cobelli E et al (2020) A reinforcement learning approach to optimal part flow management for gas turbine maintenance. Proc Inst Mech Eng Part O J Risk Reliab 234(1):52–62. https://doi.org/10.1177/1748006X19869750
Correa JCAJ, Guzman AAL (2020) Guidelines for the implementation of a predictive maintenance program. Mech Vib Condit Monit. https://doi.org/10.1016/B978-0-12-819796-7.00007-X
Correa-Jullian C, Droguett EL, Cardemil JM (2020) Operation scheduling in a solar thermal system: a reinforcement learning-based framework. Appl Energy. https://doi.org/10.1016/j.apenergy.2020.114943
Cui P, Wang J, Zhang W et al (2021) Predictive maintenance decision-making for serial production lines based on deep reinforcement learning. Comput Integrated Manuf Syst (CIMS) 27(12):3416–3428. https://doi.org/10.13196/j.cims.2021.12.004
Cui PH, Wang JQ, Li Y (2022) Data-driven modelling, analysis and improvement of multistage production systems with predictive maintenance and product quality. Int J Prod Res 60(22):6848–6865
Dahlqvist F, Patel M, Rajko A et al (2019) Growing opportunities in the Internet Of Things. https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things
Dai W, Mo Z, Luo C et al (2020) Fault diagnosis of rotating machinery based on deep reinforcement learning and reciprocal of smoothness index. IEEE Sensors J 20(15):8307–8315. https://doi.org/10.1109/JSEN.2020.2970747
Dai Z, Jiang M, Li X et al (2021) Reinforcement lion swarm optimization algorithm for tool wear prediction. In: 2021 Global reliability and prognostics and health management (PHM)—Nanjing 2021. https://doi.org/10.1109/PHM-Nanjing52125.2021.9613134
Dangut M, Jennions I, King S et al (2022) Application of deep reinforcement learning for extremely rare failure prediction in aircraft maintenance. Mech Syst Signal Process. https://doi.org/10.1016/j.ymssp.2022.108873
Das T, Gosavi A, Mahadevan S et al (1999) Solving semi-Markov decision problems using average reward reinforcement learning. Manag Sci 45(4):560–574. https://doi.org/10.1287/mnsc.45.4.560
Dau HA, Bagnall A, Kamgar K et al (2019) The UCR time series archive. Mach Learn. arXiv:1810.07758
Deloitte (2020) Industry 4.0. Deloitte Insights https://www2.deloitte.com/us/en/insights/focus/industry-4-0.html
Ding F, He Z, Zi Y et al (2008) Application of support vector machine for equipment reliability forecasting. In: 2008 6th IEEE international conference on industrial informatics, pp 526–530
Ding Y, Ma L, Ma J et al (2019) Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: a deep reinforcement learning approach. Adv Eng Inf. https://doi.org/10.1016/j.aei.2019.100977

Dogru O, Velswamy K, Ibrahim F et al (2022) Reinforcement learning approach to autonomous PID tuning. Comput Chem Eng 161(107):760. https://doi.org/10.1016/j.compchemeng.2022.107760
Dong S, Wen G, Lei Z et al (2021a) Transfer learning for bearing performance degradation assessment based on deep hierarchical features. ISA Trans 108:343–355. https://doi.org/10.1016/j.isatra.2020.09.004
Dong W, Zhao T, Wu Y (2021b) Deep reinforcement learning based preventive maintenance for wind turbines. In: 2021 IEEE 5th conference on energy internet and energy system integration (EI2), pp 2860–2865
Duan Y, Chen X, Houthooft R et al (2016) Benchmarking deep reinforcement learning for continuous control. arXiv:1604.06778
Dulac-Arnold G, Levine N, Mankowitz DJ et al (2021) Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Mach Learn. https://doi.org/10.1007/s10994-021-05961-4
Eke S, Aka-Ngnui T, Clerc G et al (2017) Characterization of the operating periods of a power transformer by clustering the dissolved gas data. In: 2017 IEEE 11th International symposium on diagnostics for electrical machines, power electronics and drives (SDEMPED), pp 298–303
Eltotongy A, Awad M, Maged S et al (2021) Fault detection and classification of machinery bearing under variable operating conditions based on wavelet transform and CNN. 2021 International Mobile. Intelligent, and Ubiquitous Computing Conference, MIUCC 2021:117–123. https://doi.org/10.1109/MIUCC52538.2021.9447673
Encapera A, Gosavi A (2017) A new reinforcement learning algorithm with fixed exploration for semi-markov control in preventive maintenance. In: ASME 2017 12th international manufacturing science and engineering conference (MSEC 2017) collocated with the JSME/ASME 2017 6th international conference on materials and processing 3. https://doi.org/10.1115/MSEC2017-2880
Epureanu B, Li X, Nassehi A et al (2020) Self-repair of smart manufacturing systems by deep reinforcement learning. CIRP Ann 69:421–424. https://doi.org/10.1016/j.cirp.2020.04.008
Erhan L, Ndubuaku M, Di Mauro M et al (2021) Smart anomaly detection in sensor systems: a multi-perspective review. In Fusion 67:64–79. https://doi.org/10.1016/j.inffus.2020.10.001
Ericsson (2021) IoT connections outlook. https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/iot-connections-outlook
Fei Y, Yang Z, Wang Z (2021) Risk-sensitive reinforcement learning with function approximation: a debiasing approach. In: International conference on machine learning (PMLR), pp 3198–3207
Feng M, Li Y (2022) Predictive maintenance decision making based on reinforcement learning in multistage production systems. IEEE Access 10:18910–18921. https://doi.org/10.1109/ACCESS.2022.3151170
Fink O, Wang Q, Svensén M et al (2020) Potential, challenges and future directions for deep learning in prognostics and health management applications. Eng Appl Artif Intell. https://doi.org/10.1016/j.engappai.2020.103678
Fons E, Dawson P, Zeng X et al (2021) Adaptive weighting scheme for automatic time-series data augmentation. arXiv preprint. arXiv:2102.08310
Frangopol DM, Lin KY, Estes AC (1997) Life-cycle cost design of deteriorating structures. J Struct Eng 123(10):1390–1401
Fujimoto S, Meger D, Precup D et al (2022) Why should I trust you, bellman? the bellman error is a poor replacement for value error. arXiv preprint. arXiv:2201.12417
Gosavi A (2004a) A reinforcement learning algorithm based on policy iteration for average reward: empirical results with yield management and convergence analysis. Mach Learn 55(1):5–29
Gosavi A (2004b) Reinforcement learning for long-run average cost. Eur J Oper Res 155(3):654–674. https://doi.org/10.1016/S0377-2217(02)00874-3
Gosavi A, Parulekar A (2016) Solving markov decision processes with downside risk adjustment. Int J Automat Comput 13(3):235–245. https://doi.org/10.1007/s11633-016-1005-3
Grzes M (2017) Reward shaping in episodic reinforcement learning. In: Proceedings of the international joint conference on autonomous agents and multiagent systems (AAMAS) 1
Hardt M, Recht B, Singer Y (2015) Train faster, generalize better: stability of stochastic gradient descent. In: Proceedings of the 33rd international conference on machinelearning
Hasselt HV, Guez A, Silver D (2016) Deep reinforcement learning with double Q-learning. In: 30th AAAI conference on artificial intelligence (AAAI 2016)
Henderson P, Islam R, Bachman P et al (2018) Deep reinforcement learning that matters. In: Proceedings of the AAAI conference on artificial intelligence, vol 32(1)
Hofmann P, Tashman Z (2020) Hidden markov models and their application for predicting failure events. Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics), vol 12139. LNCS, pp 464–477. https://doi.org/10.1007/978-3-030-50420-5_35

Hoffmann C, Altenüller T, May MC et al (2021) Simulative dispatching optimization of maintenance resources in a semiconductor use-case using reinforcement learning. In: Simulation in Produktion und Logistik 2021, Erlangen, 15–17 September 2021, p 357

Hoong Ong K, Niyato D, Yuen C (2020) Predictive maintenance for edge-based sensor networks: a deep reinforcement learning approach. In: IEEE world forum on Internet of Things (WF-IoT 2020)—symposium proceedings. https://doi.org/10.1109/WF-IoT48130.2020.9221098

Hosseinloo A, Dahleh M (2021) Deterministic policy gradient algorithms for semi-Markov decision processes. Int J Intell Syst. https://doi.org/10.1002/int.22709

Hu Q, Yue W (2003) Optimal replacement of a system according TOA semi-markov decision process in a semi-Markov environment. Optim Methods Softw 18(2):181–196

Hu Y, Miao X, Zhang J et al (2021a) Reinforcement learning-driven maintenance strategy: a novel solution for long-term aircraft maintenance decision optimization. Comput Ind Eng. https://doi.org/10.1016/j.cie.2020.107056

Hua Y, Wang X, Jin B et al (2021b) HMRL: hyper-meta learning for sparse reward reinforcement learning problem. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp 637–645

Huang J, Chang Q, Chakraborty N (2019) Machine preventive replacement policy for serial production lines based on reinforcement learning. In: IEEE international conference on automation science and engineering 2019, August, pp 523–528. https://doi.org/10.1109/COASE.2019.8843338

Huang J, Chang Q, Arinez J (2020) Deep reinforcement learning based preventive maintenance policy for serial production lines. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113701

Hui J (2021) Reinforcement learning algorithms comparison. https://jonathan-hui.medium.com/rl-reinforcement-learning-algorithms-comparison-76df90f180cf

Hutsebaut-Buysse M, Mets K, Latré S (2022) Hierarchical reinforcement learning: a survey and open research challenges. Mach Learn Knowl Extr 4(1):172–221

Icarte RT, Klassen TQ, Valenzano R et al (2022) Reward machines: exploiting reward function structure in reinforcement learning. J Artif Intell Res 73:173–208

Imagawa T, Hiraoka T, Tsuruoka Y (2022) Off-policy meta-reinforcement learning with belief-based task inference. IEEE Access 10:49494–49507

Jaakkola T, Singh S, Jordan M (1994) Reinforcement learning algorithm for partially observable markov decision problems. Adv Neural Inf Process Syst. https://proceedings.neurips.cc/paper/1994/file/1c1d4df596d01da60385f0bb17a4a9e0-Paper.pdf

Jha M, Theilliol D, Biswas G et al (2019a) Approximate q-learning approach for health aware control design. In: Conference on control and fault-tolerant systems (SysTol), pp 418–423. https://doi.org/10.1109/SYSTOL.2019.8864756

Jha M, Weber P, Theilliol D et al (2019b) A reinforcement learning approach to health aware control strategy. In: 27th Mediterranean conference on control and automation (MED 2019)—proceedings, pp 171–176. https://doi.org/10.1109/MED.2019.8798548

Kabir F, Foggo B, Yu N (2018) Data driven predictive maintenance of distribution transformers. In: 2018 China international conference on electricity distribution (CICED), pp 312–316. https://doi.org/10.1109/CICED.2018.8592417

Khan S, Farnsworth M, McWilliam R et al (2020) On the requirements of digital twin-driven autonomous maintenance. Annu Rev Control 50:13–28. https://doi.org/10.1016/j.arcontrol.2020.08.003

Knowles M, Baglee D, Wermter S (2011) Reinforcement learning for scheduling of maintenance. In: Research and development in intelligent systems XXVII: incorporating applications and innovations in Intelligent systems XVIII—AI 2010, 30th SGAI international conference on innovative techniques and applications of artificial intelligence, pp 409–422. https://doi.org/10.1007/978-0-85729-130-1_31

Kofinas P, Dounis AI (2019) Online tuning of a PID controller with a fuzzy reinforcement learning mas for flow rate control of a desalination unit. Electronics 8(2):231

Kuhnle A, Jakubik J, Lanza G (2019) Reinforcement learning for opportunistic maintenance optimization. Prod Eng 13(1):33–41

Laape S, Dollar B, Cotteleer M et al (2020) Implementing the smart factory. Deloitte Insights. https://www2.deloitte.com/us/en/insights/topics/digital-transformation/smart-factory-2-0-technology-initiatives.html

Lange S, Gabel T, Riedmiller M (2012) Batch reinforcement learning, reinforcement learning. In: Wiering M, van Otterlo M (eds) Reinforcement learning. Adaptation, learning, and optimization. Springer, Berlin, pp 45–73

Lee J, Wu F, Zhao W et al (2014) Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. Mech Syst Signal Process 42(1–2):314–334

Lepenioti K, Pertselakis M, Bousdekis A et al (2020) Machine learning for predictive and prescriptive analytics of operational data in smart manufacturing. Lecture notes in business information processing, vol 382 LNBIP, pp 5–16. https://doi.org/10.1007/978-3-030-49165-9_1

Lewis F, Vrabie D, Vamvoudakis K (2012) Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. https://ieeexplore.ieee.org/document/6315769

Li Z (2019) CWRU bearing dataset and Gearbox dataset of IEEE PHM challenge competition in 2009. https://doi.org/10.21227/g8ts-zd15

Li Z, Guo J, Zhou R (2016) Maintenance scheduling optimization based on reliability and prognostics information. In: 2016 Annual reliability and maintainability symposium (RAMS), pp 1–5. https://doi.org/10.1109/RAMS.2016.7448069

Li B, Zhou Y (2020) Multi-component maintenance optimization: an approach combining genetic algorithm and multiagent reinforcement learning. In: 2020 global reliability and prognostics and health management (PHM—Shanghai), pp 1–7

Li J, Blumenfeld DE, Huang N et al (2009) Throughput analysis of production systems: recent advances and future topics. Int J Prod Res 47(14):3823–3851. https://doi.org/10.1080/00207540701829752

Li X, Qian J, Gg Wang (2013) Fault prognostic based on hybrid method of state judgment and regression. Adv Mech Eng 5(149):562

Li Z, Zhong S, Lin L (2019) An aero-engine life-cycle maintenance policy optimization algorithm: reinforcement learning approach. Chin J Aeronaut 32(9):2133–2150. https://doi.org/10.1016/j.cja.2019.07.003

Li L, Liu J, Wei S et al (2021) Smart robot-enabled remaining useful life prediction and maintenance optimization for complex structures using artificial intelligence and machine learning. Proc SPIE. https://doi.org/10.1117/12.2589045

Lillicrap TP, Hunt JJ, Pritzel A et al (2015) Continuous control with deep reinforcement learning. arXiv e-prints. arXiv:1509.02971

Ling Z, Wang X, Qu F (2018) Reinforcement learning-based maintenance scheduling for resource constrained flow line system. In: 2018 IEEE 4th international conference on control science and systems engineering (ICCSSE 2018), pp 364–369. https://doi.org/10.1109/CCSSE.2018.8724807

Liu K, Gebraeel NZ, Shi J (2013) A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. IEEE Trans Automat Sci Eng 10(3):652–664

Liu L, Wang Z, Zhang H (2017) Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters. IEEE Trans Automat Sci Eng 14(1):299–313. https://doi.org/10.1109/TASE.2016.2517155

Liu Y, Chen Y, Jiang T (2020) Dynamic selective maintenance optimization for multi-state systems over a finite horizon: a deep reinforcement learning approach. Eur J Oper Res 283(1):166–181. https://doi.org/10.1016/j.ejor.2019.10.049

Luo Y (2021) Application of reinforcement learning algorithm model in gas path fault intelligent diagnosis of gas turbine. Comput Intell Neurosci. https://doi.org/10.1155/2021/3897077

Ma Z, Guo J, Mao S et al (2020) An interpretability research of the XGBoost algorithm in remaining useful life prediction. In: 2020 International conference on big data & artificial intelligence & software engineering (ICBASE), pp 433–438

Macek K, Endel P, Cauchi N et al (2017) Long-term predictive maintenance: a study of optimal cleaning of biomass boilers. Energy Build 150:111–117

Mahadevan S, Marchalleck N, Das TK et al (1997) Self-improving factory simulation using continuous-time average-reward reinforcement learning. In: Machine learning international workshop. Morgan Kaufmann Publishers, Los Angeles

Mahmood AR, Sutton RS, Degris T et al (2012) Tuning-free step-size adaptation. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 2121–2124

Mann L, Saxena A, Knapp GM (1995) Statistical-based or condition-based preventive maintenance? J Qual Maintenance Eng 6(5):519–541

Mao H, Liu Z, Qiu C (2021) Adaptive disassembly sequence planning for VR maintenance training via deep reinforcement learning. Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-021-08290-x

Martinez C, Perrin G, Ramasso E et al (2018) A deep reinforcement learning approach for early classification of time series. In: European signal processing conference 2018, September, pp 2030–2034. https://doi.org/10.23919/EUSIPCO.2018.8553544

Mattioli J, Perico P, Robic PO (2020) Improve total production maintenance with artificial intelligence. In: Proceedings—2020 3rd international conference on artificial intelligence for industries (AI4I 2020), pp 56–59. https://doi.org/10.1109/AI4I49448.2020.00019

Mehndiratta M, Camci E, Kayacan E (2018) Automated tuning of nonlinear model predictive controller by reinforcement learning. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3016–3021
Meng H, Ludema K (1995) Wear models and predictive equations: their form and content. Wear 181:443–457
Mikhail M, Yacout S, Ouali M (2019) Optimal preventive maintenance strategy using reinforcement learning. In: Proceedings of the international conference on industrial engineering and operations management, pp 133–141
Min W, Chao Q (2012) Reinforcement learning based maintenance scheduling for a two-machine flow line with deteriorating quality states. In: Proceedings—2012 3rd global congress on intelligent systems (GCIS 2012), pp 176–179. https://doi.org/10.1109/GCIS.2012.82
Moos J, Hansel K, Abdulsamad H et al (2022) Robust reinforcement learning: a review of foundations and recent advances. Mach Learn Knowl Extr 4(1):276–315
Morimoto J, Doya K (2005) Robust reinforcement learning. Neural Comput 17(2):335–359
Nair A, Gupta A, Dalal M et al (2020) AWAC: accelerating online reinforcement learning with offline datasets. arXiv preprint. arXiv:2006.09359
Narvekar S, Peng B, Leonetti M et al (2020) Curriculum learning for reinforcement learning domains: a framework and survey. CoRR. arXiv:2003.04960
Nectoux P, Gouriveau R, Medjaher K et al (2012) Pronostia: an experimental platform for bearings accelerated degradation tests. In: IEEE international conference on prognostics and health management (PHM'12), pp 1–8
Ng AY, Coates A, Diel M et al (2006) Autonomous inverted helicopter flight via reinforcement learning. Experimental Robotics IX pp 363–372
Ong K, Wenbo W, Friedrichs T et al (2021a) Augmented human intelligence for decision making in maintenance risk taking tasks using reinforcement learning. In: Conference proceedings—IEEE international conference on systems, man and cybernetics, pp 3114–3120. https://doi.org/10.1109/SMC52423.2021.9658936
Ong K, Wenbo W, Niyato D et al (2021b) Deep reinforcement learning based predictive maintenance model for effective resource management in industrial IoT. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3109955
Ozturk S, Fthenakis V, Faulstich S (2018) Failure modes, effects and criticality analysis for wind turbines considering climatic regions and comparing geared and direct drive wind turbines. Energies 11(9):2317
Panzer M, Bender B (2021) Deep reinforcement learning in production systems: a systematic literature review. Int J Prod Res 60(3):1–26
Paraschos P, Koulinas G, Koulouriotis D (2020) Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. J Manuf Syst 56:470–483. https://doi.org/10.1016/j.jmsy.2020.07.004
Patil S, Abbeel P (2013) Partially observable markov decision processes (POMDPs). Guest Lecture: CS287 advanced robotics
Pinciroli L, Baraldi P, Compare M et al (2020) Agent-based modeling and reinforcement learning for optimizing energy systems operation and maintenance: the pathmind solution. In: Proceedings of the 30th European safety and reliability conference and the 15th probabilistic safety assessment and management conference, pp 1476–1480. https://doi.org/10.3850/978-981-14-8593-0_5863-cd
Pinciroli L, Baraldi P, Ballabio G et al (2021) Deep reinforcement learning based on proximal policy optimization for the maintenance of a wind farm with multiple crews. Energies. https://doi.org/10.3390/en14206743
Pinciroli L, Baraldi P, Ballabio G et al (2022) Optimization of the operation and maintenance of renewable energy systems by deep reinforcement learning. Renew Energy 183:752–763. https://doi.org/10.1016/j.renene.2021.11.052
Pinto L, Davidson J, Sukthankar R et al (2017a) Robust adversarial reinforcement learning. In: International conference on machine learning (PMLR), pp 2817–2826
Plappert M, Houthooft R, Dhariwal P et al (2017b) Parameter space noise for exploration. arXiv preprint. arXiv:1706.01905
Powell WB (2009) What you should know about approximate dynamic programming. Naval Res Logist NRL) 56(3):239–249
Prashanth L, Fu MC et al (2022) Risk-sensitive reinforcement learning via policy gradient search. Found Trends Mach Learn 15(5):537–693
Prognostics HM Society (2010) 2010 PHM society conference data challenge. https://phmsociety.org/phm_competition/2010-phm-society-conference-data-challenge/

Ramasso E (2014) Investigating computational geometry for failure prognostics in presence of imprecise health indicator: results and comparisons on C-MAPSS datasets. In: PHM society European conference 2(1)

Ren Y (2021) Optimizing predictive maintenance with machine learning for reliability improvement. ASCE ASME J Risk Uncertain Eng Syst Part B Mech Eng. https://doi.org/10.1115/1.4049525

Rocchetta R, Bellani L, Compare M et al (2019) A reinforcement learning framework for optimal operation and maintenance of power grids. Appl Energy 241:291–301. https://doi.org/10.1016/j.apenergy.2019.03.027

Russenschuck S (1999) Mathematical optimization techniques. Tech. rep., CERN

Sateesh Babu G, Zhao P, Li XL (2016) Deep convolutional neural network based regression approach for estimation of remaining useful life. In: International conference on database systems for advanced applications, pp 214–228

Saxena A, Goebel K (2008) Turbofan engine degradation simulation data set. http://ti.arc.nasa.gov/project/prognostic-data-repository

Saxena A, Goebel K, Simon D et al (2008) Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 international conference on prognostics and health management, pp 1–9

Saxena A, Celaya J, Saha B et al (2010a) Evaluating prognostics performance for algorithms incorporating uncertainty estimates. In: 2010 IEEE aerospace conference, pp 1–11

Saxena A, Celaya J, Saha B et al (2010b) Metrics for offline evaluation of prognostic performance. Int J Prognost Health Manag 1(1):4–23

Saydam D, Frangopol DM (2015) Risk-based maintenance optimization of deteriorating bridges. J Struct Eng 141(4):04014120. https://doi.org/10.1061/(ASCE)ST.1943-541X.0001038

Sayyad S, Kumar S, Bongale A et al (2022) Tool wear prediction using long short-term memory variants and hybrid feature selection techniques. Int J Adv Manuf Technol 121(9):6611–6633

Schaefer AM, Udluft S, Zimmermann HG (2007) A recurrent control neural network for data efficient reinforcement learning. In: 2007 IEEE international symposium on approximate dynamic programming and reinforcement learning (IEEE), pp 151–157

Scheibelhofer P, Gleispach D, Hayderer G et al (2012) A methodology for predictive maintenance in semiconductor manufacturing. Aust J Stat 41(3):161–173

Senthil C, Pandian R (2022) Proactive maintenance model using reinforcement learning algorithm in rubber industry. Processes. https://doi.org/10.3390/pr10020371

Shen Y, Tobia MJ, Sommer T et al (2014) Risk-sensitive reinforcement learning. Neural Comput 26(7):1298–1328

Shi Y, Xiang Y, Jin T (2019) Structured maintenance policies for deteriorating transportation infrastructures: combination of maintenance types. In: Proceedings of annual reliability and maintainability symposium 2019, January. https://doi.org/10.1109/RAMS.2019.8769227

Shi Q, Lam HK, Xuan C et al (2020) Adaptive neuro-fuzzy pid controller based on twin delayed deep deterministic policy gradient algorithm. Neurocomputing 402:183–194. https://doi.org/10.1016/j.neucom.2020.03.063

Shuvo S, Yilmaz Y (2020) Predictive maintenance for increasing EV charging load in distribution power system. In: 2020 IEEE international conference on communications, control, and computing technologies for smart grids, SmartGridComm 2020 https://doi.org/10.1109/SmartGridComm47815.2020.9303021

Singh SP, Jaakkola T, Jordan MI (1994) Learning without state-estimation in partially observable Markovian decision processes. Mach Learn Proc 1994:284–292

Sinha S (2021) State of IoT 2021. https://iot-analytics.com/number-connected-iot-devices/

Skordilis E, Moghaddass R (2020) A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics. Comput Ind Eng. https://doi.org/10.1016/j.cie.2020.106600

Skydt MR, Bang M, Shaker HR (2021) A probabilistic sequence classification approach for early fault prediction in distribution grids using long short-term memory neural networks. Measurement 170(108):691

Song X, Jiang Y, Tu S et al (2019) Observational overfitting in reinforcement learning. arXiv preprint. arXiv:1912.02975

Su J, Huang J, Adams S et al (2022) Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems. Expert Syst Appl 192(116):323. https://doi.org/10.1016/j.eswa.2021.116323

Susto GA, Schirru A, Pampuri S et al (2013) A predictive maintenance system for integral type faults based on support vector machines: an application to ion implantation. In: 2013 IEEE international conference on automation science and engineering (CASE), pp 195–200

Susto GA, Wan J, Pampuri S et al (2014) An adaptive machine learning decision system for flexible predictive maintenance. In: 2014 IEEE international conference on automation science and engineering (CASE), pp 806–811

Sutton R, Barto A (2018) Reinforcement learning: an introduction, 2nd edn. MIT, Cambridge

Sutton RS, Precup D, Singh S (1999) Between mdps and semi-MDPS: a framework for temporal abstraction in reinforcement learning. Artif Intell 112(1–2):181–211

Swazinna P, Udluft S, Hein D et al (2022) Comparing model-free and model-based algorithms for offline reinforcement learning. arXiv preprint. arXiv:2201.05433

Tanimoto A (2021) Combinatorial Q-learning for condition-based infrastructure maintenance. IEEE Access 9:46788-46799. https://doi.org/10.1109/ACCESS.2021.3059244

Templier M, Paré G (2015) A framework for guiding and evaluating literature reviews. Commun Assoc Inf Syst 37(1):6

Thomas D (2020) Manufacturing machinery maintenance—NIST. National Institute of Standards and Technology (NIST), Gaithersburg. https://www.nist.gov/el/applied-economics-office/manufacturing/topics-manufacturing/manufacturing-machinery-maintenance

Thomas DS, Weiss BA (2020) Economics of manufacturing machinery maintenance. National Institute of Standards and Technology (NIST), Gaithersburg. https://doi.org/10.6028/NIST.AMS.100-34https://nvlpubs.nist.gov/nistpubs/ams/NIST.AMS.100-34.pdf

Valet A, Altenmüller T, Waschneck B et al (2022) Opportunistic maintenance scheduling with deep reinforcement learning. J Manuf Syst 64:518–534

Vogl GW, Qiao H (2021) Monitoring, diagnostics and prognostics for manufacturing operations (NIST). National Institute of Standards and Technology (NIST), Gaithersburg. https://www.nist.gov/programs-projects/monitoring-diagnostics-and-prognostics-manufacturing-operations

Walsh C (2022) Paris-Erdogan equation. https://www.maths.tcd.ie/~chas/node24.html#SECTION00841000000000000000

Wang X, Wang H, Qi C et al (2014) Reinforcement learning based predictive maintenance for a machine with multiple deteriorating yield levels. J Comput Inf Syst 10(1):9–19. https://doi.org/10.12733/jcis8124

Wang X, Qi C, Wang H et al (2015) Resilience-driven maintenance scheduling methodology for multi-agent production line system. In: Proceedings of the 2015 27th Chinese control and decision conference (CCDC 2015), pp 614–619. https://doi.org/10.1109/CCDC.2015.7161844

Wang X, Wang H, Qi C (2016) Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system. J Intell Manuf 27(2):325–333. https://doi.org/10.1007/s10845-013-0864-5

Wang H, Yan Q, Zhang S (2021a) Integrated scheduling and flexible maintenance in deteriorating multi-state single machine system using a reinforcement learning approach. Adv Eng Inf. https://doi.org/10.1016/j.aei.2021.101339

Wang X, Wang Y, Dai H (2021b) Fault diagnosis based on data-driven dynamic model. In: ICSMD 2021—2nd international conference on sensing, measurement and data analytics in the era of artificial intelligence. https://doi.org/10.1109/ICSMD53520.2021.9670767

Wang X, Xu D, Qu N et al (2021c) Predictive maintenance and sensitivity analysis for equipment with multiple quality states. Math Probl Eng. https://doi.org/10.1155/2021/4914372

Wang X, Zhang G, Li Y et al (2022) A heuristically accelerated reinforcement learning method for maintenance policy of an assembly line. J Ind Manag Optim 19(4):2381–2395

Weibull W (1951) A statistical distribution function of wide applicability. J Appl Mech 18:293–297

Weiss BA, Helu M, Vogl G et al (2016) Use case development to advance monitoring, diagnostics, and prognostics in manufacturing operations. IFAC-Papers OnLine 49:13–18. https://doi.org/10.1016/J.IFACOL.2016.12.154

Weiss BA, Alonzo D, Weinman SD (2017) Nist advanced manufacturing series 100–13 summary report on a workshop on advanced monitoring, diagnostics, and prognostics for manufacturing operations. National Institute of Standards and Technology, Gaithersburg. https://doi.org/10.6028/NIST.AMS.100-13

Wu Q, Feng Q, Ren Y et al (2021) An intelligent preventive maintenance method based on reinforcement learning for battery energy storage systems. IEEE Trans Ind Inf. https://doi.org/10.1109/TII.2021.3066257

Xanthopoulos A, Kiatipis A, Koulouriotis D et al (2017) Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system. IEEE Access 6:576–588. https://doi.org/10.1109/ACCESS.2017.2771827

Yan S, Ma B, Zheng C et al (2019) An optimal lubrication oil replacement method based on selected oil field data. IEEE Access 7:92110–92118. https://doi.org/10.1109/ACCESS.2019.2927426

Yang D (2022) Adaptive risk-based life-cycle management for large-scale structures using deep reinforcement learning and surrogate modeling. J Eng Mech. https://doi.org/10.1061/(ASCE)EM.1943-7889.0002028

Yang Z, Qi C (2013) Preventive maintenance of a multi-yield deteriorating machine: using reinforcement learning. Syst Eng Theory Pract 33(7):1647–1653

Yang H, Shen L, Cheng M et al (2018) Integrated optimization of scheduling and maintenance in multi-state production systems with deterioration effects. Comput Integr Manuf Syst (CIMS) 24(1):80–88. https://doi.org/10.13196/j.cims.2018.01.008

Yang H, Li W, Wang B (2021) Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning. Reliab Eng Syst Saf. https://doi.org/10.1016/j.ress.2021.107713

Zhang N, Si W (2020) Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. Reliab Eng Syst Saf. https://doi.org/10.1016/j.ress.2020.107094

Zhang Z, Tang Q (2022) Integrating preventive maintenance to two-stage assembly flow shop scheduling: Milp model, constructive heuristics and meta-heuristics. Flexible Serv Manuf J 34(1):156–203. https://doi.org/10.1007/s10696-021-09403-0

Zhang C, Vinyals O, Munos R et al (2018) A study on overfitting in deep reinforcement learning. arXiv:1804.06893

Zhang C, Gupta C, Farahat A et al (2019) Equipment health indicator learning using deep reinforcement learning. Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics), vol 11053. LNAI, pp 488–504. https://doi.org/10.1007/978-3-030-10997-4_30

Zhang P, Zhu X, Xie M (2021) A model-based reinforcement learning approach for maintenance optimization of degrading systems in a large state space. Comput Ind Eng. https://doi.org/10.1016/j.cie.2021.107622

Zheng S, Ristovski K, Farahat A et al (2017a) Long short-term memory network for remaining useful life estimation. In: 2017 IEEE international conference on prognostics and health management (ICPHM), pp 88–95

Zheng S, Ristovski K, Farahat A et al (2017b) Long short-term memory network for remaining useful life estimation. In: 2017 IEEE international conference on prognostics and health management (ICPHM), pp 88–95

Zheng W, Lei Y, Chang Q (2017c) Reinforcement learning based real-time control policy for two-machine-one-buffer production system. In: ASME 2017 12th international manufacturing science and engineering conference, MSEC 2017 collocated with the JSME/ASME 2017 6th international conference on materials and processing 3. https://doi.org/10.1115/MSEC2017-2771

Zonta T, da Costa C, da Rosa Righi R et al (2020) Predictive maintenance in the industry 4.0: A systematic literature review. Comput Ind Eng. https://doi.org/10.1016/j.cie.2020.106889