Deep-Reinforcement-Learning-Based Predictive Maintenance Model for Effective Resource Management in Industrial IoT

Kevin Shen Hoong Ong, *Member, IEEE*, Wenbo Wang, *Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, and Thomas Friedrichs

***Abstract*—Unplanned breakdown of critical equipment interrupts production throughput in Industrial IoT (IIoT), and data-driven predictive maintenance (PdM) becomes increasingly important for companies seeking a competitive business advantage. Manufacturers, however, are constantly faced with the onerous challenge of manually allocating suitably competent manpower resources in the event of an unexpected machine breakdown. Furthermore, human error has a negative rippling impact on both overall equipment downtime and production schedules. In this article, we formulate the complex resource management problem as a resource optimization problem to determine if a model-free deep reinforcement learning (DRL)-based PdM framework can be used to automatically learn an optimal decision policy from a stochastic environment. Unlike the existing PdM frameworks, our approach considers PdM sensor information and the resources of both physical equipment and human as part of the optimization problem. The proposed DRL-based framework and proximal policy optimization long short term memory (PPO-LSTM) model are evaluated alongside baselines results from human participants using a maintenance repair simulator. Empirical results indicate that our PPO-LSTM efficiently learns the optimal decision-policy for the resource management problem, outperforming comparable DRL methods and human participants by 53% and 65%, respectively. Overall, the simulation results corroborate the proposed DRL-based PdM framework's superiority in terms of convergence efficiency, simulation performance, and flexibility.**

***Index Terms*—Decision-support systems, deep reinforcement learning (DRL), Industrial Internet of Things (IIoT), predictive maintenance (PdM), resource management.**

Manuscript received June 17, 2021; revised August 16, 2021; accepted August 22, 2021. Date of publication September 3, 2021; date of current version March 24, 2022. This work was supported in part by the Collaboration Program between Computer Networks and Communications Lab, Nanyang Technological University of Singapore and Robert Bosch (SEA) Pte Ltd.; in part by the Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), National Research Foundation, Singapore, through AI Singapore Programme (AISG) under Award AISG-GC-2019-003; in part by WASP/NTU under Grant M4082187 (4080); and in part by the Singapore Ministry of Education (MOE) Tier 1 under Grant RG16/20. *(Corresponding author: Wenbo Wang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by University's Institutional Review Board (IRB).

Kevin Shen Hoong Ong and Dusit Niyato are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ongs0129@ntu.edu.sg; dniyato@ntu.edu.sg).

Wenbo Wang is with the Faculty of Engineering, Bar Ilan University, Ramat Gan 5290002, Israel (e-mail: wangwen@biu.ac.il).

Thomas Friedrichs is with the IT Strategy and Innovation Asia Pacific, Robert Bosch (SEA) Pte Ltd., Singapore 573943 (e-mail: thomas.friedrichs@sg.bosch.com).

Digital Object Identifier 10.1109/JIOT.2021.3109955

I. Introduction

INFREQUENT equipment maintenance results in erratic production of defective goods, wastes resources, and results in significant revenue losses. Predictive maintenance (PdM), enabled by the Industrial Internet of Things (IIoT), seeks to minimize unscheduled downtime of equipment through early identification of potential failures from online sensor data. However, owing to the heterogeneity of manufacturing equipment, maintenance expense, and resource constraints, establishing a generic PdM framework that learns to manage resources (e.g., physical equipment and human) remains a major challenge. Therefore, PdM-based resource management is a promising maintenance strategy for advancing toward a generic PdM framework [1].

According to [2], the majority of existing works use deep learning (DL) techniques in equipment failure prognostics. For example, DL-based solutions are used to improve equipment remaining useful life (RUL) estimation and anomaly detection [3]–[6]. Some researchers have recently examined the applicability of deep reinforcement learning (DRL) to PdM, with promising findings. To diagnose and classify faults in time-series-based equipment health sensor data, Martinez *et al.* [7] and Ding *et al.* [8] proposed using deep Q network (DQN). Similarly, [9] investigates the use of temporal difference (TD) Learning for RUL forecasting. Finally, [10] proposes the use of Double DQN to learn the optimal replacement point for equipment based on its health index value, with good generalization performance across similar equipment. But PdM encompasses more than equipment maintenance [1]. Instead, considerable emphasis should also be placed on enabling maintenance automation to optimize maintenance strategy, particularly in the area of human resource management.

Motivated by this research gap, the improved PdM framework must also consider human resource management, ensuring that maintenance tasks are allocated to the most competent technician. Although [11]–[13] make specific references to PdM and human resource management in their work, their assumptions are mainly limited to simpler and ideal maintenance scenarios. Note that the highlighted papers represent a fraction of the PdM-related works, and we refer readers to surveys [1]–[3] for details. Ultimately, both businesses and decision makers will greatly benefit from the data-driven

maintenance recommendations, since they will be able to take the best possible action for any physical equipment.

To address the stochastic resource optimization problem, we propose a DRL-based model framework that leverages the proximal policy optimization (PPO) long short term memory (LSTM) (i.e., PPO-LSTM) model. First, we introduce the edge-powered PdM framework architecture, which enables PdM for a network of equipment within a generic production facility. Second, we formulate pertinent PdM and resource management elements into a Markov decision process (MDP) framework to discover the optimal decision policy. Furthermore, we coin the term "equipment severity rating," which quantifies the probability of equipment failure in relation to the widely used equipment health indicator value in PdM literature. Third, a model-free PPO-LSTM model is presented to address the reward sparsity issue in stochastic settings and discover the optimal decision policy given the stochastic resource optimization problem. Specifically, the LSTM module is used to capture pertinent spatial-temporal information before further processing by the PPO model. Fourth, we conduct extensive simulation experiments using a maintenance repair simulator (MRS) to train the PPO-LSTM agent on the relationship between the equipment severity rating, the maintenance cost model, and the technician competence level. Additionally, we undertake IRB-approved real-world experiments with two groups of working professionals to provide a human-level benchmark against which performance is compared to. Consequently, we demonstrate the efficacy of our proposed approach as a decision-support tool. Finally, our DRL approach is also extended to the NASA C-MAPSS data set, a well-known time-series PdM data set, and observed positive results. Our *main contributions* in this article are as follows.

1) We formulate the PdM manpower allocation into a resource optimization problem and present a DRL-based PdM Model Framework. The overall optimization objective is to increase production revenues while maximizing the cumulative equipment uptime in an IIoT network powered by edge computing. With the proposed data-driven framework in place, the routine but challenging task of manpower resource allocation is now automated, resulting in increased productivity for both production and maintenance teams.
2) We propose the PPO-LSTM model for automating the decision-making process associated with PdM-based resource management. LSTM is used to improve the performance of PPO in stochastic settings, while PPO is responsible for selecting the best state action to perform.
3) We perform extensive simulation experiments using an MRS to evaluate the performance of PPO-LSTM, and we undertake IRB-approved real-world experiments, comprised of working professionals, to provide a human-level baseline for performance comparison. Empirically, the proposed PPO-LSTM outperforms both comparable DRL methods and human participants by 53% and 65%, respectively.
4) We also expand the PPO-LSTM model to the NASA C-MAPSS data set, a well-known time-series PdM data set. Interestingly, PPO alone reports a 73% improvement in learning efficiency relative to prior work. Overall, these empirical results demonstrate the effectiveness of our proposed DRL-based PdM Maintenance Model framework as a decision-support tool.

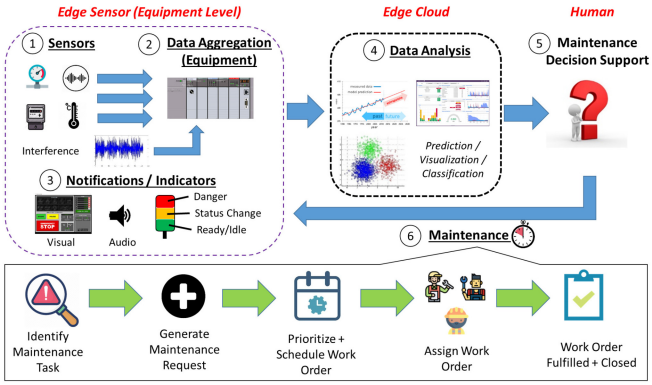

Fig. 1. PdM model overview: equipment data generation (steps 1 and 2), equipment data notification and indicators (step 3), equipment data analysis (step 4), maintenance decision-support (step 5), and maintenance task workflow (step 6).

II. Preliminary and Related Work

In this article, PdM serves two purposes, namely, equipment health assessment, and maintenance resource management. In this section, we briefly describe each function and summarize the symbiotic relationship in Fig. 1. Thereafter, we describe and critically analyze the limitations of related work and existing PdM-based system models to motivate our research question.

A. Equipment Health Assessment

In situ sensors in modern manufacturing equipment monitor numerous process parameters for signs of process deviations that can affect product quality and provide early warning of equipment failure, see Fig. 1 (steps 1 and 2). However, numerous factors can corrupt sensor data, resulting in anomalous data, in the form of false system alarms and alert notifications, see Fig. 1 (step 3). An anomaly is a statistical concept that refers to data points which vary greatly from other measurements, and previous studies [14]–[17] suggest that the fault evolution of various system degradation models may be modeled as an exponential decay function following:

$$F(t) = e^{-\lambda t}. \tag{1}$$

In addition, it is proposed in [18] that the initial degradation conditions, stress, and wear rate of individual components can also be generalized into a normalized time-varying health index (H_t) with the following mathematical expression:

$$H_t = 1 - g - e^{at^b} \tag{2}$$

where the initial degradation condition of g is nonzero; a and b are generalized wear rates that correspond to the effects of temperature and relevant subsystem stress terms [18].

Overall, the health index models the various component subsystems, and the trained machine-learning model can be deployed on the equipment either in situ or retrofitting of edge sensors (ESs) [10]. At the same time, decoupling the model training and retraining processes needed for each equipment type opens up a conceptual paradigm shift that significantly reduces the risk of retraining a monolithic machine-learning model. In this work, we reformulate the time-varying health index in [10] to better reflect the maintenance context, with formal definitions described in Section IV-A.

B. Equipment Maintenance Resource Management

One of PdM's primary functions is to forecast the RUL, which may actually lower the economic cost of maintenance. When the cost model-derived maintenance interval T_i is less than the equipment's RUL T_{RUL}, the cost model result may be used for maintenance decision making. On the contrary, $T_i > T_{\text{RUL}}$ implies that equipment failure is impending prior to the next monitoring point. Therefore, maintenance decisions are made using RUL data received from the PdM model, as shown in Fig. 1 (step 4).

The proposed secondary function of PdM is the allocation and management of the maintenance resources, see Fig. 1 (steps 5 and 6). Maintenance resources, in this sense, refer to a group of maintenance staff comprising of engineers and technicians. Effective management of maintenance resources is considered an NP-hard combinatorial problem [19], and conventional mathematical programming techniques are incapable of providing exact solutions within a reasonable time frame. Consider the conventional maintenance routine, where varying complexity of maintenance tasks are allocated to a fixed shift of technicians with different responsibilities. Simultaneously, the maintenance engineer is accountable for manually allocating maintenance tasks to suitably competent technicians. In this regard, the maintenance engineer may easily have neglected other maintenance factors, such as the cost of the technician assigned, complexity of equipment maintenance, and expected mean time to repair. As a result, more equipment downtime is unintentionally incurred relative to the first effort at assigning competent maintenance technicians. Due to the fast-paced nature of the work environment, technicians may perform internal *ad hoc* task rescheduling, which increases the likelihood of longer-than-planned equipment downtime.

C. Machine Learning for PdM Applications

Deep-Learning: The broad applications and impacts of Deep-Learning can be found in industry applications, such as RUL estimation of aircraft engine [16], RUL of ball bearings [5], RUL of battery life [14], signal feature extraction [6], and anomaly detection [20]. Some commonly used models include convolutional neural network (CNN), recurrent neural network (RNN), deep belief networks (DBNs), and auto encoder (AE). These works are often targeted on specific equipment components diagnosis, and no existing work has attempted to extend their proposed solution to multicomponent systems, atypical of the real-world equipment [1].

Reinforcement Learning (RL): DRL continues to be appealing to companies since it eliminates the labor-intensive data labeling process, and is slowly gaining traction in PdM applications. In [8], the fault diagnosis problem is formulated as a diagnostic game, which is then applied to roller bearings and hydraulic pumps with very high classification accuracy results achieved. Similarly, the work in [7] utilizes DQN for data classification with the UCR data set [21]. The study in [9] suggests the use of TD learning to derive the equipment health state degradation, and is evaluated on the NASA C-MAPSS data set. From these examples, DRL usage appears to have been mostly limited to fault classifications and RUL estimation until recently. Ong *et al.* [10] formulated the equipment health indicator as a function of equipment run time with the aim of recommending an appropriate replacement point based on the equipment's health. Their proposed Double-DQN-based solution is evaluated using the NASA C-MAPSS data set, and has been observed to achieve good learning performance and sampling efficiency. Despite these encouraging results, DRL-based solutions for PdM applications remain largely under-explored.

Maintenance Recommendation Systems and Framework: Reference [11] propose a general IoT-based framework for businesses to utilize when selecting appropriate fleet management systems (FMSs) platform solutions for real-world use. SERENA [22] is a platform for predictive analytic that runs on a lightweight hybrid architecture that combines cloud and edge computing. The SERENA-enabled predictive analytic service, however, provides only equipment condition monitoring using a variety of established machine learning algorithms: Random Forest, Decision Tree, and Gradient Boosted Tree. On the contrary, [23], [24] focus on actionable maintenance recommendations. Both works, which are based on the PHM 2013 Data Challenge, utilize collaborative filtering and Bayesian inference methodology to achieve accurate RUL estimate results for maintenance recommendation. Wen *et al.* [4] focused on the RUL estimation and proposes the use of Genetic Programming to fuse data from various sensor sources into a composite Health Index, as stated in (2). More recently, performance degradation modeling and threshold-based monitoring approach are investigated in [25] to alert users proactively. However, their approach suffers from a high incidence of false alarms as the human resource component is ignored. As a result, Teoh *et al.* [12] proposed the use of the genetic algorithm (GA) for asset management, which encompasses both the machine and human component, in order to reduce equipment failure. Unfortunately, GA-based solutions may not scale well with increasing problem complexity, learning convergence may not be as efficient as other optimization methods, and the solution may be inferior to human-level performance. Ong *et al.* [13] examined the potential of DRL for decision-support maintenance management and compares the performance to those of working professionals. In this article, we extend the findings in [13] by considering additional maintenance and human resource parameters in our proposed PdM Model Framework.

TABLE I
COMPARISON OF PDM-BASED RESOURCE MANAGEMENT METHODS WITH EXISTING WORKS

References	Resource		Predictive Equipment Maintenance	Fog / Cloud / Edge / Local	Resource Performance Metrics				Method
	Physical	Human			Time	Cost	Competency Levels	Maintenance Complexity	
[16]	✓	-	✓	Local	-	-	-	-	Attention-based LSTM
[5]	✓	-	✓	N/A	-	-	-	-	LSTM
[14]	✓	-	✓	N/A	-	-	-	-	Deep Belief Network
[6]	✓	-	✓	Local	-	-	-	-	Stacked AutoEncoder
[20]	✓	-	✓	N/A	-	-	-	-	Cumulative Sum-based LSTM
[23]	✓	-	✓	N/A	-	-	-	-	Collaborative Filtering
[24]	✓	-	✓	N/A	-	-	-	-	Bayesian Networks
[7], [8]	✓	-	✓	N/A	-	-	-	-	DQN
[4]	✓	-	✓	Local	-	-	-	-	Genetic Programming
[9]	✓	-	✓	N/A	-	-	-	-	Temporal Difference Learning
[10]	✓	-	✓	Edge	-	-	-	-	DDQN-PER-PN
[11]	✓	✓	-	N/A	-	-	-	-	IoT Asset & Fleet Management Framework
[22]	✓	-	✓	Fog	-	-	-	-	Random Forest, Decision Tree, Gradient Boosted Tree
[26]	✓	✓	✓	N/A	-	-	-	-	Maintenance 4.0 System Architecture
[12]	✓	✓	✓	Fog	✓	✓	-	-	Genetic Algorithm
[13]	✓	✓	✓	Local	✓	✓	1	-	Recurrent Advantage Actor-Critic
Proposed work	✓	✓	✓	**Edge**	✓	✓	**3**	✓	**PPO, PPO-LSTM and DRL-based PdM Framework**

D. Critical Analysis

PdM enables maintenance team to address issues prior to equipment failure, and Table I summarizes the existing PdM and resource management techniques. As IIoT-enabled PdM is still in its infancy [2], the majority of existing work focuses solely on improving the RUL estimation accuracy and ignores the manpower resource management [27], [28]. Except for [13], none of the existing work addresses all four performances metrics: 1) time; 2) cost; 3) competency levels; and 4) maintenance complexity when proposing a resource management method that encompasses both physical and human resources. For example, operational and maintenance costs, such as skilled technician's man hours, mean time to repair, parts replacement, and maintenance budget, are often overlooked in existing PdM framework. Moreover, [13] considers only a Junior technician level with a constant repair probability, while the proposed work takes into account several levels of competence and varying repair probabilities.

Given the aforementioned issues, it is essential to investigate and propose an alternative PdM framework. Notably, the research question that our paper address is: how to manage effectively the symbiotic relationship between the technicians, equipment, and AI in a complex environment using sequential decision-making methods while simultaneously optimize revenue as a function of production performance under uncertainty. In this article, the proposed PdM framework can include complementary AI-based decision support to facilitate effective resource management across both physical and human resources. In addition, the four performance metrics should be taken into consideration. The IIoT sensor data is fully utilized for the PdM model training, and the utilization of edge-based sensors will alleviate IIoT network congestion. Owing to these features, this article addresses the challenges of the existing resource management technique, and further details are described in Section III.

III. SYSTEM MODEL AND PDM FRAMEWORK

To put Fig. 1 into perspective, we consider a generic production facility for predictive equipment maintenance in IIoT (Fig. 2), which comprises an edge cloud (EC), ES, and manpower resources. At the equipment level, the system model consists of a network of ESs with direct connection to the EC. Due to computational resource constraints, each ES aggregates in situ time-series sensor data and transmits it to EC for storage and analysis, as shown in Fig. 1 (steps 1 and 2). Meanwhile, a predictive model (i.e., agent) monitors the incoming data for anomalous behavior and triggers notifications or alarms based on a preset parameter threshold.

A. Proposed Predictive Maintenance Framework Architecture

Design Objective: The pandemic's widespread effects continues to be felt as businesses struggle to remain financially viable, and low-wage employees like technicians are perceived as replaceable by automation or rehiring. Given the commercial confidentiality on staffing capacity information, we will assume in this article that the maintenance personnel to critical equipment ratio increases from 1:1 to 1:10. In particular, our proposed framework considers the real-world constraints of the maintenance budget and technician actions within the decision-making framework, which are both challenging to model and under-explored in similar literature. The proposed framework is briefly illustrated in Fig. 2 for reference purposes.

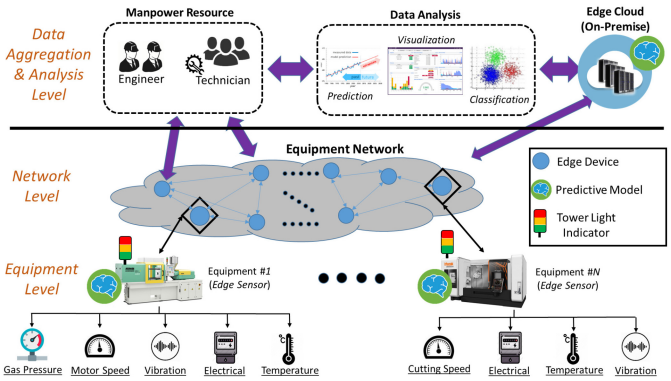

Fig. 2. System overview of the proposed PdM framework for IIoT networks with edge computing capability per equipment, with on-device predictive models.

Edge Sensor (Equipment): A static ES symbolizes an intelligent sensor predictive model that generates the metadata of the monitored equipment. Metadata is considered to be an abstract representation of the complex relationship between multiple components, where sensors constantly monitor essential component parameters. Examples of sensor data include pump pressure, oil temperature, and milling speed. Generally, a condition-based approach establishes the minimum and maximum limits for each sensor in order to estimate the current health state of the equipment on the basis of domain knowledge. Such an approach is likely to result in a significant number of false alarms, reduces the overall equipment runtime, and wastes precious network bandwidth. Alternatively, an in situ PdM model may be deployed on an ES device with adequate computational resource, and generates metadata upon detection of abnormal sensor signals, such as probabilistic-based alarm information and maintenance action recommendations, whilst preserving precious network bandwidth.

Here, we build on [10]'s work, and the proposed list of available actions is based on the data from multiple sensors observed: [*Severity*, *Repair*, *Replace*, *Hold*]. *Severity* denotes the suspected abnormality pertaining to the operational state of the equipment, raw sensor data inputs, and the severity ratings can either be software or hardware based. Depending on the severity level, the PdM model may recommend either *Repair* or *Replace* to the maintenance team. With adequate data and model training, the PdM model is able to assess the optimal action to be taken via a threshold-free approach, and can be remotely deployed on the ES device in order to achieve the outset objective. To be clear, both *Severity* Level 1 and *Hold* are the default states and actions. Besides, the generated metadata is transmitted to the data aggregation and analysis (DAA) layer via a network of ES network (ESN), which can either be a wired or wireless network. The ESDs need only communicate with the EC with minimal network latency following the use of high-speed communication technologies, such as Gigabit Ethernet, Wi-Fi 6 (IEEE 802.11ax), and 5G.

Edge Cloud (On-Premise): In general, the on-premise EC has orders of magnitude more processing power and memory resources than ES. EC offers several functionalities: 1) storage for industrial big data; 2) machine learning pipeline analytic, such as data preprocessing, model training and testing, prediction, and model deployment; 3) real-time monitoring and analysis of production planning; and 4) resource management for application-specific decision-making. These resources include but are not limited to human, autonomous robots, and departments. In this article, we are only concerned with the human manpower resource.

Manpower Resource: Recent survey [29] highlights that a wide variety of competent technicians are expected to retire within the next decade, causing a wider gap in labor's skill/knowledge. To close the gap and achieve effective resource management, we propose that the manpower resource layer in Fig. 2 offers multiple functions: 1) database of relevant employee information, such as experience levels; 2) estimated manpower cost; and 3) behavioral risk profile. We focus on the issues of maintenance management, which are essential to optimizing production time of the machinery. One of the decision-making priorities is to assign the correct maintenance technician, for any given equipment severity rating, in order to maximize the probability of fixing the equipment at the first time round, thereby minimizing the equipment downtime. Furthermore, each technician wears a mobile wireless device that functions as a PdM model feedback mechanism, and the user inputs are captured by means of feedback ratings through the mobile device's touch-screen interface.

IV. Problem Formulation

Our main objective is to optimize the total production throughput with minimum cost by leveraging the current manpower resource. Given the cost constraints, achieving such tradeoff is challenging. We first consider the ESN[1] and DAA layers, and formalize their objectives separately. Then, we merge both layers' definitions and formulate the two-layer interactions. Finally, we conduct problem transformation to contextualize the system parameters and obtain the optimal task assignment strategy. For readers' convenience, we provide a summary of notations used across this article in Table II.

A. ESN—Sensor MetaData

The objective of the ESN is defined in (3), with the aim to maximize the cumulative uptime (ρ_E) of a network of equipment, where g_q denotes the individual uptime of equipment q ($q \in \{1, 2, \ldots, N\}$). Without loss of generality, a single ES is assumed and communicates directly with the on-premise EC

$$\max \rho_E = \sum_{q=1}^{N} g_q. \tag{3}$$

Random ambient noise can affect sensor measurements, and a higher incident rate of false alarms is to be anticipated if the noise characteristics are not quantifiable or observable.

[1]This model was introduced in [10] with simple assumptions, which does not consider the significant implications of severity rating in a resource management problem within the broader maintenance framework. Furthermore, the use of raw sensor data with an unknown noise function necessitates the use of a different problem formulation, which is not considered in [10], either.

TABLE II
LIST OF IMPORTANT NOTATIONS

Symbol	Definition
ρ_{E}	(Objective) Maximise equipment network uptime
ρ_{D}	(Objective) Optimise resource management
ρ_{H}	(Objective) Optimise user-rating
g_q	Equipment q's uptime
ι	Overall Factory Revenue
N	Number of equipment
M	Number of manpower resource (e.g. technician) set
Γ	Manpower cost w.r.t. experience-level set
τ	Maintenance Budget
$\mathcal{C}$	Cost constraint w.r.t. Maintenance Action
U	User-ratings received w.r.t. contextual situation set
x_t^i	ith sensor's data value at t time-step
H_t	Normalised equipment health value at t time-step
ψ, ψ_t^i	Normalised Equipment Severity Rating set, scalar value with i ratings at time-step t
$\mathcal{S}_{\mathrm{E}}^{\psi}$, $\mathcal{S}_D^{\psi}$	(State) Discrete severity level at ESN and DAA layers respectively
S^Z	(State) Human emotion
n	Number of human emotion categories/levels
Υ	Equipment priority
χ	Array or group of repair actions
κ	(Action) Idle/Hold
η	(Action) Repair
ϵ	(Action) Replace
y_η	Number of repair actions taken until successful fix
y_ϵ	Number of replace actions taken until successful fix
ω_{k}	Positive constants
π^*	Optimal policy
α	Learning Rate
γ	Discounting factor
$\hat{A}_t$	Estimation of Advantage Function at t time-step
G	Number of actors in actor-critic network
d_{target}	Target value of KL Divergence
β	Weighted factor for KL-Divergence
f_t, i_t, o_t	Forget, Input and Output Gates at t time-step
h_t	State of hidden layer at t time-step
b_f, b_i, b_C, b_o	Bias vectors
$c_t, \tilde{c}_t$	Cell state memory, Cell state candidate at t time-step
σ	Sigmoid activation function

Considering external noise as a hidden feature and ES data acquisition is Markovian [9], we propose modeling the complex environment as a sequential decision-making problem using the partially observable MDP (POMDP) framework. Within the ESN layer, the PdM module is denoted as an agent for modeling purposes.

Sensor ($\mathcal{S}_E^x$): At each equipment, an agent observes at every time step t the sensor state information, denoted by x_t^i, where $i \in \{1, 2, \ldots, \varrho\}$; ϱ denotes the upper-bound constraint on the number of sensors per equipment for monitoring and practicality purposes. Considering that the sensor data (i.e., state) is sampled at every time step, the cumulative states theoretically become a continuous state space, and providing exact solutions within a reasonable time period becomes impractical [30]. We handle this limitation by grouping the data samples into a series of time slices (φ) of constant length μ. Namely, each φ value is a discrete representation of the arithmetic mean sensor data ($\bar{x}_t^i$), such that $\bar{x}_t^i \in \{x_1^i, x_2^i, \ldots, x_\mu^i\}$. Without loss of generality, $\bar{x}_t^i$ is formalized within the environmental state ($\mathcal{S}_E^x$) as $\mathcal{S}_E^x \leftarrow \bar{x}_t^i \ \forall\varphi$.

Operating Condition ($\mathcal{S}_E^L$): Owing to the repeated use of equipment and complex environmental conditions, the overall performance of the equipment steadily deteriorates over time and a concave-like exponential decay trend [18] is observed before eventual equipment failure, see Fig. 3(a). Likewise, the sensor's life expectancy decreases over time where some sensors fail faster than others due to ageing and environmental exposure, such as operating temperature. For modeling purposes, the environmental state ($\mathcal{S}_E^L$) can be influenced by the operating temperature condition (L) and binary operating status is utilized. For instance, normal operating status ($\mathcal{S}_E^L = 0$) is user defined and conditional on $L \in [25, 70]$. Otherwise, abnormal operating status ($\mathcal{S}_E^L = 1$) is assumed.

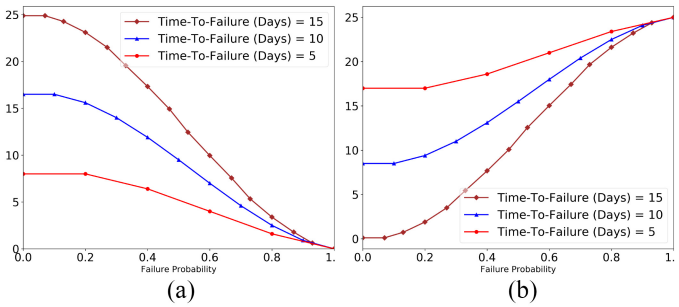

Fig. 3. H_t in (2) is obtained by dimension reduction of the multiple sensor data acquired using Principal Component Analysis, and follows the decay pattern in (1). We then slice H_t over an arbitrary time interval (in days) and apply the Markov chain rule to obtain the state-transition values as H_t approaches 0, indicative of equipment failure. The corresponding range of severity ratings is based on (7). (a) Relative equipment health. (b) Relative severity.

Severity Rating ($\mathcal{S}_E^{\psi}$): Given N equipment, each ESN-level equipment outputs a severity rating (ψ_t^i), where $i \in [0, \delta]$ at every time step (t), and the range of severity rating is arbitrarily defined. We normalize i, where $\delta = 1$ and ψ_t^i is formalized within environment state ($\mathcal{S}_E^{\psi}$) as $\mathcal{S}_E^{\psi} \leftarrow \psi_t^i$. The observed severity rating sequentially increases in accordance to the equipment's operational status. For example, we can assume that $\psi_t^{0.2}$ indicates normal operational state while ψ_t^1 indicates critical operational state.

Thus, the overall ESN-based state space $s \in \mathcal{S}_E$ is summarized as an N-set Cartesian product using our system model and POMDP framework as follows:

$$\mathcal{S}_E = \mathcal{S}_E^x \times \mathcal{S}_E^L \times \mathcal{S}_E^{\psi}. \tag{4}$$

Action ($\mathcal{A}_E$): The action space comprises of *Hold*(κ), *Repair*(η), and *Replace*(ϵ), and a maintenance budget (τ) is managed by the agent. In order to simulate the maintenance decision-making process, an action-based cost constraint ($\mathcal{C}$) helps to ensure the agent establishes a reasonable maintenance strategy, where $\mathcal{C} \in \{\mathcal{C}_\kappa, \mathcal{C}_\eta, \mathcal{C}_\epsilon\}$ actions are available. By assuming the equipment's health degradation follows (1), the agent is then tasked with selecting the sequence of actions to take at each state without violating τ. By default, κ action incurs zero maintenance cost and the list of action cost relationship is defined as follows:

$$\mathcal{A}_E = \begin{cases} (\epsilon, \eta, \kappa) \Big| \\ \sum_{q=1}^{N} \mathcal{C}_\epsilon^q \geq 2 \sum_{q=1}^{N} \mathcal{C}_\eta^q \text{ and } \beta - \sum_{q=1}^{N} \mathcal{C}_\epsilon^q \geq 0 \\ \sum_{q=1}^{N} \mathcal{C}_\epsilon^q \leq \sum_{q=1}^{N} (\mathcal{C}_\eta^q / 2) \text{ and } \beta - \sum_{q=1}^{N} \mathcal{C}_\eta^q \geq 0. \end{cases} \tag{5}$$

Belief State Transition ($\mathcal{O}$): Compared to a new sensor, the damaged sensor can record values of the sensor state values that are likely different, and such behavior is detected externally due to multiple state skipping. Assuming that the multiple state skipping phenomenon can be used to infer the existence of an indirectly observable interference, the environmental state (s) is therefore deemed partially observable (o). For convenience, we set o to be s, and the observation-transition probability ($\mathcal{O}$) is instead used to identify the true value of s. In a sense, the environmental state in POMDP is encoded, and the agent relies on a set of *beliefs* (b) for a probability distribution over s, where $b_t(s) = P(s_t = s|o_t, a_{t-1}, \ldots, b_o)$ and b_o is the initial belief vector. The Bayes rule is then used to calculate the belief state transitions, defined as follows:

$$b'(s) = P(s|o, a, b) = \frac{\Omega\big(o|s', a\big)\sum_{s\in S} T\big(s'|s, a\big)b(s)}{\sum_{s'\in S}\Omega(o|s', a)\sum_{s\in S} T(s'|s, a)b(s)}. \quad (6)$$

For an equipment under monitoring, the belief state transitions for each equipment sensor are iteratively calculated and analyzed by the agent. A single set of complementary metadata information is comprised of *Equipment Health* (H) and *Severity* (ψ), defined in (2) and (7), respectively. Recall, H presumably follows the exponential decay trend in (2). Intuitively, we expect ψ to increase with decreasing values of H_t. Formally, $\psi \in \psi_t^i$ is a relative complement of $H \in H_t$, where $H_t \in [0, 1]$ and $\psi \in [0, 1]$. For reference, we visually describe the complementary relationship in Fig. 3(b) and is mathematically expressed as follows:

$$\psi = 1 - H. \quad (7)$$

Given the stochastic environment, the agent will randomly choose actions from (5) to perform based on the observed belief state changes and transitions. Consequently, an equipment's H_t can be restored with ϵ to an almost new condition, while η regresses H_t to a previously observed state by y_η states. Furthermore, the quality of repairs varies and the equipment health state change $\phi(\mathcal{S}_E)$ will differ depending on the $\mathcal{A}_E$. For example, performing η at $\mathcal{S}_E = y-1$ induces a belief state transition from $\mathcal{S}_E = y$ to $\mathcal{S}'_E = y - |\phi(\mathcal{S}_E)|$. Such behavior is concisely represented as $y_\eta \in \{\phi(\mathcal{S}_E)|\mathcal{S}_E \in \mathcal{A}_E\}$.

Reward ($\mathcal{R}_E$): To contextualise (3) within the POMDP framework, we propose for ρ_E be re-expressed as $\mathcal{R}_E$, where $\rho_E \rightarrow \mathcal{R}_E$. Based on the observed values of $\mathcal{A}_E$ and $\mathcal{S}_E$, the agent learns to make better decisions, and this behavior is motivated by the following reward function:

$$r_t(s_t, a_t) = \begin{cases} \mathcal{R}_\epsilon, & \text{if } \mathcal{S}_E^x > 0, \ \beta > 0 \\ \mathcal{R}_\eta, & \text{if } \mathcal{S}_E^x > 0, \ \beta > 0 \\ \mathcal{R}_{\text{Exp}}, & \text{if } \mathcal{S}_E^x > 0 \\ \mathcal{R}_{\text{Frug}}, & \text{if } \mathcal{S}_E^x > 0, \ \beta > 0 \\ -1, & \text{Otherwise} \end{cases} \quad (8)$$

where $r_t \in \mathcal{R}_E$, $s_t \in \mathcal{S}_E$, and $a_t \in \mathcal{A}_E$ at every time step (t). Given current state s_t and $(\tau - \mathcal{C}_\epsilon > 0) \wedge (\tau - \mathcal{C}_\eta > 0)$, the agent selects either Replace (r_ϵ) or Repair (r_η) actions and values of s_{t+1} and r_t are received. Otherwise, the agent is penalised where $(s_t = H_t = 0) \vee (a_t = \kappa)$. In order to mitigate the reward sparsity problem in real-world applications, the agent is encouraged to explore the problem state space using $\mathcal{R}_{\text{Exp}}$. Similarly, $\mathcal{R}_{\text{Frug}}$ is defined to positively reinforce the agent's frugal behavior when learning the optimal action to take, emulating real-world human decision-making parameter.

B. DAA—Resource Management

Similar to ESN in (3), the DAA layer (ρ_D) aims to optimize the total equipment run time (g_q), based on the equipment severity ratings (ψ) and maintenance budget constraints (τ), and is denoted as follows:

$$\max \rho_D = \max \rho_E \Big| (\psi, \tau) = \max \sum_{q=1}^{N} g_q \Big| (\psi, \tau). \quad (9)$$

Given the similarity to ESN's objective, and that DAA's environmental constraints are fully observable, we exploit this information and re-express the objective function as a fully observable MDP. Within the DAA layer, the resource management model or human participant is abstracted as an agent for modeling purposes. Every equipment is uniquely represented as $\mathcal{S}_E^i$, where $1 \leq i \leq N$ denotes the ith equipment in the equipment network.

Equipment Severity Rating ($\mathcal{S}_D^\psi$): Previously in Section IV-A, the initial values of $\mathcal{S}_E^\psi$ are considered to be partial observations o in the POMDP context. Subsequently, the o values are believed to be accurate because data processing using traditional machine learning technique is utilized to calculate and acquire accurate equipment health H metadata as defined in (7). As a result, the o values of ψ in (7) can be regarded as the true equipment severity rating state within the fully observable MDP context. In other words, based on the received severity rating information from $\mathcal{S}_E^\psi$ in (4), the observed data is assumed accurate and requires no further data processing. Thus, we can mathematically represent $\mathcal{S}_E^\psi$ within DAA's environment state ($\mathcal{S}_D^\psi$) as $\mathcal{S}_D^\psi \leftarrow \mathcal{S}_E^\psi$.

User Emotion ($\mathcal{S}_D^Z$): Unlike existing works that assume a completely rational agent, the human action is a function of multiple parameters, such as emotional states, age, and risk-aversion attitude [31]. While the human emotional states are stochastic, we employ the Markov chain [32], [33] to model human emotions ($\mathcal{S}_D^Z$) into n emotional states, such as *Calm*, *Cautious*, and *High Alert* with conditional constraints in (10). In addition, other factors may affect the equipment's severity level to behave stochastically with the similar behavior as observed over N equipment

$$\mathcal{S}_D^Z = \begin{cases} \text{Calm}, & \text{if } Z \in \big[0, 1/(n)\big]\big|\{n \in \mathbb{R}\} \\ \text{Cautious}, & \text{if } Z \in \big[0.01 + 1/n, 2/n\big]\big|\{n \in \mathbb{R}\} \\ \text{High Alert}, & \text{if } Z \in \big[0.01 + 2/n, 1.0\big]\big|\{n \in \mathbb{R}\}. \end{cases} \quad (10)$$

With reference to our system model and MDP framework, we integrate the $\mathcal{S}_D^Z$ into the DAA-based state space $s \in \mathcal{S}_D$ as an N-set Cartesian product as follows:

$$\mathcal{S}_D = \mathcal{S}_D^{(1,\,\psi,\,Z)} \times \mathcal{S}_D^{(2,\,\psi,\,Z)} \times \cdots \times \mathcal{S}_D^{(N,\,\psi,\,Z)} \quad (11)$$

where $s \in \mathcal{S}_D$; $N \in [1, \infty]$ represents the equipment index; $Z \in [0, 1]$ represents the normalized emotional state; n denotes the user-defined levels of human emotional states; anf ψ denotes the equipment's fully observable severity level state.

Although the degradation behavior of no two identical equipment is alike, continuous equipment usage, after some time, inherently leads to an exponential increase in occurrences of nonoperational states, where $\psi_t^i > 1$. In the event where $N > 1$ equipment reports $\psi_t^i > 1$ values, a human operator psychologically associates and combines present state information [34], prioritizes the equipment information received, before focusing on the subsequent course of action.

Action ($\mathcal{A}_D$): The similarities between $\mathcal{A}_E$ and the proposed action space ($\mathcal{A}_D$) is $\mathcal{A}_E \subset \mathcal{A}_\mathrm{D}$, where $\mathcal{A}_\mathrm{D}$ comprises two additional independent action groups: 1) *Equipment Priority* (Υ) and 2) *Repair Type* (χ). An array of $\mathcal{S}_\mathrm{D}^{(N,\,i,\,Z)}$ values, from (11) are stored within Υ, and *heuristic* is used to recommend the appropriate equipment sequence to action upon. The scalar actions of *Hold*(κ), *Repair*(η), and *Replace*(ϵ) can be compacted as $\chi \in \{\kappa, \eta, \epsilon\}$. The frequencies of actions ϵ and η are finite, constraint by τ, so as to imitate real-world decision-making. We propose to characterize the maintenance action constraint as $\mathcal{C}$, where $\mathcal{C} \in \{\mathcal{C}_\kappa, \mathcal{C}_\eta, \mathcal{C}_\epsilon\}$ actions are available. Otherwise, κ remains the default action and zero cost is incurred. Labor costs (Γ) in particular take into account the aforementioned maintenance action as well as the skill levels of the dispatched technician. For example, the skill levels can be classified as: 0 to 5 years ($\Gamma_{l=1}$), 6 to 15 years ($\Gamma_{l=2}$), and $\geq$15 years ($\Gamma_{l=3}$), where $\Gamma \in \Gamma_l \mid \Gamma_l \in \{\Gamma_1, \Gamma_2, \Gamma_3\}$. The action space, which includes the corresponding maintenance action and manpower costs, can then be defined as follows:

$$\mathcal{A}_D = \begin{cases} \left(\overbrace{\epsilon, \eta, \kappa}^{\chi}, \Gamma, \Upsilon \right) \mid \\ \sum_{q=1}^{N} \mathcal{C}_\epsilon^q \geq 2 \sum_{q=1}^{N} \mathcal{C}_\eta^q \text{ and } \tau - \Gamma - \sum_{q=1}^{N} \mathcal{C}_\epsilon^q \geq 0 \\ \sum_{q=1}^{N} \mathcal{C}_\epsilon^q \leq \sum_{q=1}^{N} \left(\mathcal{C}_\eta^q / 2 \right) \text{ and } \tau - \Gamma - \sum_{q=1}^{N} \mathcal{C}_\eta^q \geq 0. \end{cases} \tag{12}$$

State-Action Transition ($\mathcal{T}$): Consider that the following $N = 3$ equipment states are observed: $\mathcal{S}_D^{(1,\,0.75)}$, $\mathcal{S}_D^{(2,\,1)}$, $\mathcal{S}_\mathrm{D}^{(3,\,0.25)}$. The optimization algorithm first chooses action Υ and heuristically determines the equipment order priority as: $\mathcal{S}_D^{(2,\,1)}$, $\mathcal{S}_\mathrm{D}^{(1,\,0.75)}$. Thereafter, the agent will determine an appropriate action to perform, based on the current value i, and notify the appropriate maintenance technician accordingly. However, in the real world, the maintenance technician is unable to perform maintenance on multiple equipment at the same time, and the severity rating of each unattended equipment will remain at current levels, with κ continuously invoked, until the maintenance personnel invokes either ϵ or η. For example, ϵ is performed on $N = 2$ equipment index, and the observed changes in severity rating state ($\phi(\mathcal{S}_\mathrm{D})$) transitions from $\mathcal{S}_\mathrm{D}^{(2,\,1)}$ to $\mathcal{S}_\mathrm{D}^{(2,\,0.25)}$, which is the default equipment severity rating state, see Fig. 4.

Likewise, η generally reverts the current severity rating state toward $\mathcal{S}_D^{(1,\,0.25)}$. Besides, repair quality is likely to vary, and additional χ actions may be required to adjust an equipment's

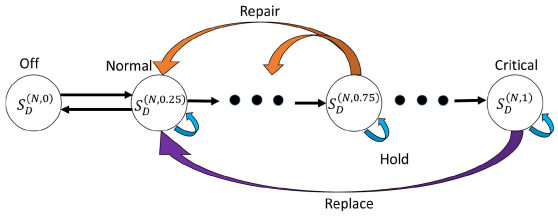

Fig. 4. Example of severity rating state transition for Nth equipment w.r.t. types of maintenance action repair performed by the maintenance agent (e.g., human technician or optimization algorithm).

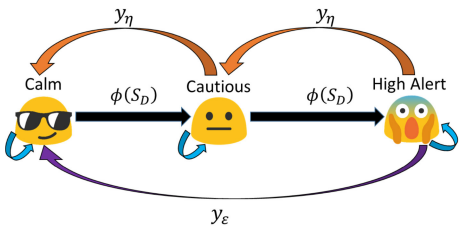

Fig. 5. Proposed human emotional state-transition for maintenance resources based on emotional and mental state transition network models [37], [38] in response to external stimuli [36], such as equipment severity rating.

$\phi(\mathcal{S}_D)$ by y_η times. For reader's convenience, we summarize the aforementioned state transitions in (13) and (14), respectively

$$y_\epsilon \in \phi(\mathcal{S}_D), \text{ if } \mathcal{S}_\mathrm{D}^{(N,i)} \in \chi, 0.75 \leq i \leq 1 \tag{13}$$

$$y_\eta \in \phi(\mathcal{S}_D), \text{ if } \mathcal{S}_D^{(N,i)} \in \chi, 0.25 < i \leq 1. \tag{14}$$

According to [31], the Wundt curve model [35] is widely used to model the underlying human behavioral trend with respect to increasing rewards and increasing stimulus intensity. Notably, it is possible to reinterpret the Wundt curve model by splitting it into two partial systems, i.e., a primary reward system and a risk-aversion system with respect to increasing external stimuli [35]. Intuitively, we can correlate external stimuli with equipment severity rating and behavioral-based human actions with emotional states. Thus, we can use the state-action transition diagram in Fig. 5 to visually describe these correlations, and the actions under consideration includes both action groups Υ and χ. Sudhof *et al.* [36] empirically validated the plausibility that human emotions undergo temporal state transitions that typically follow exponential decay, with the decay rate also being situational dependent. In the absence of relevant literature for our maintenance-based research problem, we propose following [36]'s state-transition assumptions and evaluate it against the state transitions from real-world human participant experimental data.

Considering the maintenance context problem, we assume a nonlinear correlation exists between $\mathcal{S}_D^\psi$ and $\mathcal{S}_D^Z$ for N equipment. When, for example, the severity rating state of equipment index $N = 1$ rises from $\mathcal{S}_\mathrm{D}^{(1,\,\psi=0.2)}$ to $\mathcal{S}_\mathrm{D}^{(1,\,\psi=0.3)}$, a similar user emotional transition is predicted following (10). Given $Z \in \mathcal{S}_\mathrm{D}^Z \mid \{n = 3\}$, no emotional state transition occurs until $\psi > 1/n$, which consequently triggers $\mathcal{S}_\mathrm{D}^Z$ state transition from *Calm* to *Cautious*, as shown in Figs. 4 and 5, respectively.

Formally, we can express this state-transition correlation as

$$\phi(\mathcal{S}_D) \Rightarrow \left\{\phi\left(\mathcal{S}^{\mathrm{Z}}\right) | Z \in \chi,\ Z \in [0,1]\right\}. \tag{15}$$

Reward ($\mathcal{R}_D$): When invoking an action from $\mathcal{A}$ given severity rating of state $\mathcal{S}_D$, the quality of the decision-making policy can be improve via the reward function $\mathcal{R}_D(s_t, a_t, s_{t+1})$. Recall (9), the cumulative sum of equipment runtime can be redefined as the cumulative rewards received from N equipment, based on the values of $\mathcal{S}$ and $\mathcal{A}$ taken at each time step. In consideration of the time-variant repair action, we let the success probability of the η action $P(\varsigma_j = 1 \mid \varsigma_{j-1} = 0)$ be uniformly distributed within Ω time steps. Hence, a successful repair is denoted as $\varsigma_j = 1$ and the reward function is defined as follows:

$$r_t(s_t, a_t) = \begin{cases} r_\epsilon, & \text{if } \xi \wedge a_t = \epsilon \\ r_\eta = \Omega - j + 5, & \text{if } \xi \wedge a_t = \eta \wedge \varsigma_j = 1 \\ r_{\eta-1} = +5, & \text{if } \xi \wedge a_t = \eta \wedge \\ & \left(0 < \varsigma_j < 1\right) \\ -0.05, & \text{Otherwise} \end{cases} \tag{16}$$

where $r_t \in \mathcal{R}_D$ at every time step (t) and $\xi \Rightarrow (\mathcal{S}_D^{\psi} > 0, \tau > 0)$; $j \in [1, \ldots, \Omega]$ represents time-to-repair, defining at which time step the equipment repair is successful. Given the current value of $\mathcal{S}_D$, the reward signal from r_ϵ and r_η corresponds to the Replace and Repair actions, respectively. Furthermore, we enforce a negative reward to encourage state-space exploration at every time step regardless of the state-action pair selection. For the purpose of contextualizing (9) within the MDP framework, we re-express ρ_D as the maintenance resource reward function $\mathcal{R}_D$, where $\rho_D \rightarrow \mathcal{R}_D$.

C. DAA—User-Rating

Acquiring user ratings is often challenging, and the analytical process is complicated by the lack of dependent variables. As in the maintenance scenario, we define the user ratings ($U \in U_p$) as an interplay of multiple situational factors, and the user-ratings function (ρ_U) is defined as follows:

$$\max \rho_U = U_p \times \psi_t^i \times \Gamma_l \tag{17}$$

where Γ_l and ψ_t^i refer to the skill level of the technician and the equipment severity rating, respectively. In other words, both ψ_t^i and Γ_l can be loosely represented as state $s \in \mathcal{S}_U \mid \{\mathcal{S}_U \leftarrow \psi_t^i\}$ and action $a \in \mathcal{A}_U \mid \{\mathcal{A}_U \leftarrow \Gamma_l\}$. M represents the total number of equipment technicians and the user-defined user-ratings are normalize to $U_p \in [0, 1]$, where $p \in [1, 2, \ldots, M]$. For example, let us consider a user-rating range between 0 (worst) to 10 (best), and the user 2 inputs a feedback rating of 8. Therefore, $U_p = 0.8$ (i.e., [8/10]).

Next, we cast ρ_U into the MDP framework by re-expressing $\rho_U \rightarrow \mathcal{R}_U$, where $\mathcal{R}_U$ denotes the reward function for user-ratings. Consequently, an optimization algorithm (i.e., user-rating agent) learns to select the optimal action ($\mathcal{A}_U$) with respect to the $\mathcal{S}_U$ in order to maximize $\mathcal{R}_U$ received, which generally leads to improved values of U_p.

D. Overall Problem Formulation

Recall, the aim of this article is to collectively maximize overall factory revenue (ι) while optimizing the factory resources, as described in Sections IV-A–IV-C. Taking these multiple objectives into account, the subjunctives be integrated into the MDP Framework as: $\mathcal{S} \in \{\mathcal{S}_E,\ \mathcal{S}_D,\ \mathcal{S}_U\}$, $\mathcal{A} \in \{\mathcal{A}_E,\ \mathcal{A}_D,\ \mathcal{A}_U\}$, and $\mathcal{R} \in \{\mathcal{R}_E,\ \mathcal{R}_D,\ \mathcal{R}_U\}$.

Considering the optimization problem for resource management, the resource management algorithm (i.e., agent) interacts with the environment state $\mathcal{S}$ at t time step, selects an action $\mathcal{A}$ to perform, and receives new values of $\mathcal{S}$ and $\mathcal{R}$ from the environment, respectively. As a result, the agent will continue to interact with the environment iteratively, optimizing the actions taken based on each state value in order to maximize the cumulative rewards received $\mathcal{R}$. In retrospect, we let $\mathcal{R} \in \mathcal{R}_\iota$ by re-expressing $\iota \rightarrow \mathcal{R}_\iota$, and the new factory revenue reward function is designed as follows:

$$\begin{aligned} \max\ & \mathcal{R}_\iota = (\omega_1 \mathcal{R}_E + \omega_2 \mathcal{R}_D + \omega_3 \mathcal{R}_U) \mid \mathcal{C} \\ \text{s.t.}\ & (a) : \omega \in \{\omega_1, \omega_2, \omega_3\} \\ & (b) : \omega_1 + \omega_2 + \omega_3 = 1 \\ & (c) : \mathcal{C} \in \left\{\mathcal{C}_\kappa, \mathcal{C}_\eta, \mathcal{C}_\epsilon\right\} \end{aligned} \tag{18}$$

where $\mathcal{C}$ refers to the maintenance action cost constraints from (12), and the remaining parameter constraints are defined in (18) (a)–(c). Positive weight constants $\omega_1, \omega_2,$ and ω_3 are necessary for balancing the three subrewards. For page economy and research scope reasons, we set $\omega_1 = 0.1$, ω_2=0.9, $\omega_3 = 0$, where ω_3 is considered for future work.

V. Problem Transformation Based on RL

The model optimization problem in (18) is challenging to solve as the optimization objective requires to manage maintenance resource effectively in the absence of limited or lack of information for modeling purposes. Moreover, the selection of an insufficiently skilled technician to conduct maintenance repair on an equipment with critical severity status potentially leads to a suboptimal solution (i.e., revenue reduction due to longer equipment downtime). Likewise, learning a hidden Markov model from noisy or stochastic environments is challenging for the RL agent, particularly when incomplete and temporal-dependent environment information are critical to obtaining the optimal solution of the model optimization objective. In this context, traditional model-based dynamic programming tools are also unsuitable due to the significance of time-critical maintenance and an agent's inability to anticipate what the next state will be before the chosen action is taken.

Therefore, in this section, we apply the MDP framework to address our maintenance resource management problem and adopt model-free RL as a solution tool. In what follows, we describe how the MDP framework can be used to achieve optimal decision-making policy. For clarity and brevity, standard RL notations are used for the parameters of *state*, *action*, and *rewards*. The limitations of related RL approaches are briefly highlighted to motivate our proposed DRL solution.

The RL agent's objective is to learn an optimal policy from (18) through *trial-and-error* interactions within the stochastic environment. For every interaction with the environment, the agent receives information about the next state s_{t+1} in addition to the current state reward r_t received. Then, the agent attempts to maximize the long-term cumulative expected reward values of being in state s_t recursively, and the optimal state-value policy ($V_*(s)$) is achieved by maximizing the value function, in (19), over all existing decision policies, where $\gamma \in [0, 1]$ is the discounting factor

$$V_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma r_{t+1} | s_t = s\right]. \tag{19}$$

The state transition probability $P(s', r|(s, a))$ of any stochastic environment is both dynamic and unknown. Hence, the RL agent's strategy is to search recursively for an optimal decision policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ that maps the state $s_t \in \mathcal{S}$ to action $a_t \in \mathcal{A}$. The Q-learning algorithm can learn the optimal policy by maximizing the action-value function ($Q_\pi(\mathcal{S}, \mathcal{A})$) over all Q-value policies in (20). The Q-value is recursively updated using TD Learning [30], and the off-policy transitions (s_t, a_t, r_t, s_{t+1}) is learnt

$$Q_\pi(s, a) = \mathbb{E}_\pi\left[r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a\right] \tag{20}$$

$$Q^*(s, a) = (1 - \alpha)Q(s, a) + \alpha Q_{\text{obs}}(s, a). \tag{21}$$

The optimal Q-function is obtainable as $Q^*(s, a) = \max_\pi V_\pi(s, a)$. The Q-function is updated following (21), where α is the learning rate and $Q_{\text{obs}}(s, a) = r(s, a) + \gamma \max_{a' \in \mathcal{A}} Q'(s', a')$. With $Q^*(s, a)$, the optimal policy is

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} .Q^*(s, a). \tag{22}$$

Reward sparsity is a well-known RL problem and with policy gradient-based RL methods, the effect of multiple actions makes it challenging to identify the series of optimal actions to take given a sequence of steps/states, especially in an online setting. In the context of our maintenance problem, the state-of-the-art actor–critic RL approach is more adept and is able to converge to an optimal decision policy in both online and offline policy settings. For instance, the generalized advantage estimator (GAE) [39] approach calculates the advantage of taking an action by using a weighted average of individual advantages over n-steps so as to reduce the variance of the estimator whilst minimizing bias. However, the use of multiple actors with GAE tradeoff learning efficiency with high variance in subsequent policy network update intervals [40], which can result in suboptimal decision policy convergence too. A potential solution would be to integrate a shared LSTM module into the actor–critic network in order to reduce policy network variance in estimating actions based on the current environment state, at the expense of increased GPU memory resource requirements and longer training time.

In this work, we propose using the PPO method to improve the policy network variances of actor–critic solutions, which is responsible for action estimation and is further discussed in the next section. Briefly, PPO imposes penalty-like restrictions to further reduce the variance between policy network updates, at the expense of adding some bias while reducing the occurrence of the agent taking suboptimal actions. Likely benefits include quicker learning convergence and attaining optimal performance for our maintenance resource management problem. Recently, Furthermore, we would also like to investigate whether it is possible for a nonmemory-based actor–critic solution to outperform the LSTM-based actor–critic solutions before extending the comparison to even more challenging equipment maintenance problems.

VI. Proximal Policy Optimization for Effective Maintenance Resource Management

Policy gradient methods operate by calculating an estimation of a policy gradient and optimizing it by using the stochastic gradient ascent algorithm. The estimator $\hat{g}$ can be obtained by differentiating the objective

$$\mathcal{L}^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t|s_t)\hat{A}_t\right] \tag{23}$$

where $\hat{\mathbb{E}}_t[\cdots]$ denotes an empirical average expectation of a batch of finite samples within a sampling and optimization algorithm; π_θ denotes a stochastic policy and $\hat{A}_t$ is an estimation of the advantage function at time step t.

Policy gradient methods suffer from two main problems: 1) *Unstable Policy Updates* and 2) *Data Inefficiency* [30]. Policy changes are unpredictable for policy gradient methods because of their large step updates leading to poor policy updates, which consequently lead to learning bad policies. On the contrary, smaller step updates lead to slower learning. It is also preferable for these learning methods to learn from recent experience and exploit. However, current policy gradient methods discard this experience following gradient changes and this exacerbates the learning process, as a neural network requires a large amount of data to learn effectively. In this section, we propose that these issues be mitigated through the PPO algorithm [40]. Furthermore, to cope with stochastic environments with long-term dependencies, we suggest supplementing PPO with LSTM, which we explain in this section.

A. Objective Clipping

Based on the idea of importance sampling and a neural network's preference for normalized data, PPO requires to maintain two policy networks. The first policy network $\pi_\theta(a_t|st)$ is used to refine the policy updates based on the previous policy $\pi_{\theta_{\text{old}}}(a_t|st)$, in which this ratio is clipped and the minimum of the both policy actions will instead be considered. In doing so, large policy network updates will be restricted by the clipping threshold (ϵ), and the clipped objective function is described as follows:

$$\mathcal{L}^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\right)\hat{A}_t\right] \tag{24}$$

where the probability ratio $r_t(\theta) = [\pi_\theta(a_t|s_t)/\pi_{\theta_{\text{old}}}(a_t|s_t)]\hat{A}_t$ and $r_t(\theta_{\text{old}}) = 1$. Depending on the value of $\hat{A}_t$, the choice of clipping ratio, $\text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)$, can be either $1 - \epsilon$ or $1 + \epsilon$ interval range. The pseudocode is shown in Algorithm 1.

Algorithm 1 PPO With Clipped Objective

Input: Initial policy parameters θ_0, clipping threshold ϵ
for k=0,1,2,... **do**
  Collect set of partial trajectories D_k on policy $\pi_k = \pi(\theta_k)$
  Estimate advantages $\hat{A}_t^{\pi_k}$ using GAE algorithm [39]
  Compute policy update:
    $\theta_{k+1} = argmax_\theta \, L_{\theta_k}^{CLIP}(\theta)$
  by taking K steps of minibatch SGD (via Adam), where
  $\mathcal{L}_{\theta_k}^{CLIP}(\theta)$
  $= \hat{\mathbb{E}}_t\Big[\min\Big(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\Big)\hat{A}_t\Big]$
end for

Algorithm 2 PPO With Adaptive KL Penalty Coefficient

Input: Initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ
for k=0,1,2,... **do**
  Collect set of partial trajectories D_k on policy $\pi_k = \pi(\theta_k)$
  Estimate advantages $\hat{A}_t^{\pi_k}$ using generalised advantage estimation algorithm
  Compute policy update:
    $\theta_{k+1} = argmax L_{\theta_k}(\theta) - \beta_k D_{KL}(\theta||\theta_k)$
  by taking K steps of minibatch SGD (via Adam)
  if $D_{KL}(\theta_{k+1}||\theta_k \geq 1.5\delta)$ **then** $\beta_{k+1} = 2\beta_k$
  else if $D_{KL}(\theta_{k+1}||\theta_k \leq \delta/1.5)$ **then** $\beta_{k+1} = \beta_k/2$
  end if
end for

B. Adaptive Kullback–Liebler Penalty Coefficient

With reference to trusted region policy optimization (TRPO) [41], we can assume that the optimal policies calculated in the trust region are always better, with some upper bound guarantee, over the old policy. Thus, the objective function can be calculated as

$$\max_\theta \; \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t\right] - \beta KL\big[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\big] \quad (25)$$

where β induces a weighted factor to the Kullback–Liebler (KL) KL-divergence penalty, to penalize or incentivize some target value of KL-divergence (d_{target}) during policy updating. In other words, the KL-penalised objective, after several policy updates by stochastic gradient descent, can be written as

$$\mathcal{L}^{KLPEN}(\theta) = \hat{\mathbb{E}}_t\big[r_t(\theta) - \beta KL\big[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\big]\big]. \quad (26)$$

Likewise, the KL-divergence, denoted as d in (27), is also computed after every policy updates such that if $d < d_{\text{targ}}/1.5, \beta \leftarrow \beta/2$; $d > d_{\text{targ}} \times 1.5, \beta \leftarrow \beta \times 2$. Then, the updated β value is used in the next policy update interval

$$\hat{\mathbb{E}}_t\big[r_t(\theta) - \beta KL\big[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\big]\big]. \quad (27)$$

As a result, PPO is able to inherit TRPO performance, and is optimized by gradient descent methods. For reference, the pseudocode algorithm is described in Algorithm 2.

C. Recurrent Neural Network

LSTM [42] is a variant of RNN, and is often used in DRL literature for spatial-temporal feature learning. Individual cells can extract feature states across a recurrent network while preserving temporal information within each cell state, and LSTM uses a gated-like structure for selective transmission of sequential information. Each LSTM cell comprises of forget gates f_t, input gates i_t, and output gates o_t. The gate structures are formally described as follows:

$$\begin{aligned} f_t &= \sigma_l\big(W_f \cdot \big[h_{t-1}, x_t\big] + b_f\big) \\ i_t &= \sigma_l\big(W_i \cdot \big[h_{t-1}, x_t\big] + b_i\big) \\ \tilde{c}_t &= \tanh\big(W_c \cdot \big[h_{t-1}, x_t\big] + b_c\big) \\ c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t \\ o_t &= \sigma\big(W_o \cdot \big[h_{t-1}, x_t\big] + b_o\big) \\ h_t &= o_t * \tanh(c_t). \end{aligned} \quad (28)$$

From (28), we denote W_f, W_i, W_c, and W_o as weight matrices and b_f, b_i, b_c, and b_o as bias vectors for input vector of sensor data x_t at time step t; c_t denotes the cell state memory at t time step, h_{t-1} represents the state of the hidden layer at time step $t-1$ whereas h_t represents the state of the hidden layer at time step t; $\tilde{c}_t$ represents a candidate for some cell state at time step t; $*$ denotes the element-wise multiplication of the vectors and the gated structure behavior follows a sigmoid activation function σ.

D. PPO Algorithm

For our maintenance simulation problem, we consider an image-based state space (i.e., 500 × 500 pixels) with state representation at the pixel level. Therefore, a CNN is warranted for PPO's policy and value function to learn via shared parameters, hereby termed PPO-CNN. In addition, a loss function is required to backpropagate the policy target and value function gradients for optimization purposes. In PPO, we also consider an entropy bonus S to manage the state-space exploration and exploitation tradeoff in a similar way to the epsilon-greedy strategy of DQN. Following [40], S can be combined with (23), (24), and (26) at each iteration to optimize an overall objective function, defined as follow:

$$\mathcal{L}^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\Big[\mathcal{L}^{CLIP}(\theta) - c_1\mathcal{L}^{VF} + c_2 S[\pi_\theta](s_t)\Big]. \quad (29)$$

From (29) we denote c_1, and c_2 as regularization coefficients, S denotes an entropy bonus; $\mathcal{L}^{VF}$ is a compact representation of the squared error loss between the learned state-value function $V(s)$ and the target state-value function as $(V_\theta(s_t) - V_t^{\text{target}})^2$.

One major drawback of parameter sharing, of both value and policy networks, is performance instability due to the simultaneous backpropagation of gradients across the network during the model learning phase. As such, we plan to empirically identify suitable hyperparameters to manage this issue. In this work, the PPO algorithm to be implemented utilizes a fixed-length trajectory where G actors will each collect T time steps of training data in parallel. Then, the losses for

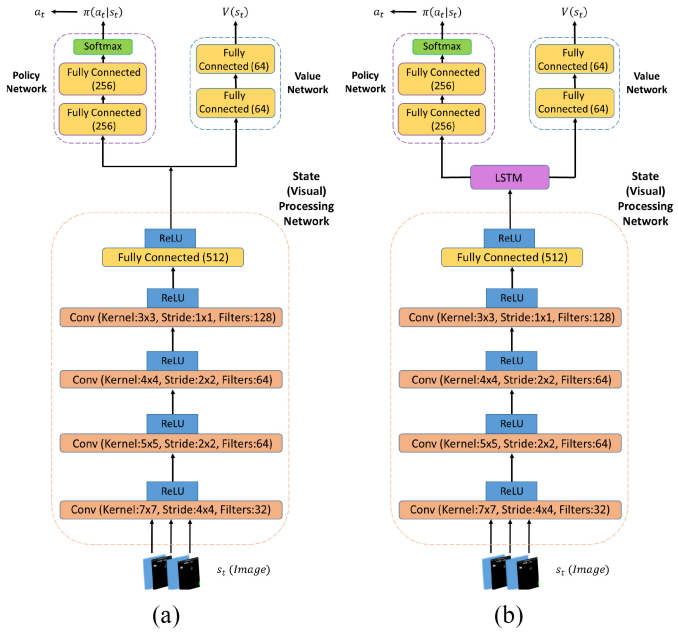

Fig. 6. PPO-based actor–critic architecture variants. (a) CNN-based PPO (PPO-CNN). (b) CNN+LSTM-based PPO (PPO-CNNLSTM).

each corresponding objectives on GT time steps of data will be optimized using a gradient descent-based methods for K epochs, such as the Adaptive Moment Estimation Optimizer (Adam).

The above-mentioned solution, however, is suitable for stochastic environments with short spatial-temporal dependencies (i.e., state-action value pair) and may not achieve optimal results, as it requires the PPO agent to retrieve older state-action sequence information. For example, given the same equipment and technician, the mean-time-to-repair (MTTR) of the equipment can vary greatly due to the different rate of equipment degradation and environmental factors. To address this issue, we propose to modify existing PPO-CNN architecture by inserting an LSTM layer between the Convolutional and Feedforward layers. This model variant is termed PPO-LSTM [Fig. 6(b)], and the cells in the LSTM layer with $h_t = \text{LSTM}(o_t,\ h_{t-1})$ are thus used to estimate the $Q(h_t, a_t)$ instead of $Q(s_t, a_t)$. To be clear, o_t refers to the current observation o_t and may not necessarily correspond to the current environment state s_t while h_{t-1} represents the state of the hidden layer at time step $t-1$.

In summary, the proposed policy network constraints of *objective clipping*, *adaptive KL divergence penalty,* and *entropy bonus* will be validated with an actor–critic-based neural network solution. For reader's understanding, the proposed actor–critic-based PPO architecture variants are shown in Fig. 6 and the pseudocode is given in Algorithm 3.

VII. Experimental Setup

Due to the scope of the proposed DRL framework, we propose and describe three experiments in this section.

Algorithm 3 PPO, Actor–Critic Style

1: **for** iteration=1,2,... **do**
2: **for** actor=1,2,...,N **do**
3: Run policy $\pi_{\theta_{\text{old}}}$ in environment for T time-steps
4: Compute advantage estimates $\hat{A}_1, \ldots, A \hat{+} t$
end for
5: Optimise surrogate L w.r.t. θ, with K epochs and minibatch size $M \leq GT$
6: $\theta_{\text{old}} \leftarrow \theta$
end for

A. Maintenance Repair Simulator

We create an MRS that is adequately versatile for a range of purposes, such as model training and validation for the DRL agent and data collection from human participants for benchmark purposes. As a baseline, we set $\Omega = 120$ in MRS and at each simulation step, the machine repair success probability decreases linearly as $(1/120)$, $(1/119)$ at $i = 2$, and so on to emulate real-world scenario in which repair times differ with varying equipment severity rating. Similarly, we set five severity levels and generate three technician skill levels (e.g., junior, senior, and expert). Through iterative interaction with MRS, the rewards received by the DRL model also takes into account the different cost constraints and will, based on the skill level of the selected technician, learn an optimal decision-making policy to satisfy (18). We then report the corresponding results for all severity levels, technician skill levels, and PPO, respectively.

B. Human Participants

A total of 26 working professionals participated in our IRB-approved experiment which consists of both white-collar and blue-collar workers. The mix-gender participants, aged between 20 and 50 are from Singapore and China, and the overall average participant age falls within the range of 30 to 40 years old. Each participant is presented with a set of instruction and the game objective, which is to maximize the total game rewards received. Relevant game data is automatically captured and every human participant utilizes the keyboard to navigate the MRS game environment. Additionally, a warm-up game followed by two games is permitted for each participant, where each game comprises of 30 rounds of gameplay, for example. Yet unbeknownst to all participants, they will play the same game environment under four different game difficulty, where game difficulty is synonymous with equipment severity rating.

The experimental data for all participants is aggregated, preprocessed, and we perform the statistical analysis. Example of data collected for each human participant are user's score, time taken to game completion, and a snapshot of actions taken to complete each game. Thereafter, we demonstrate the efficacy of the AI-based solution for maintenance decision making by comparing the human participant results to the proposed DRL variants. For disclaimer purposes, the anonymity of all human participant is strictly enforced for data security and privacy purposes throughout the course of experiments.

TABLE III
C-MAPSS DATA SET[2] UNDER TEST

Dataset	FD001	FD003
Training Set	100	100
Test Set	100	100
Operating Conditions	1	1
Fault Conditions	1	2

C. Turbofan Engine Data Set and Data Preparation

The NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) data set [18] is generated from a commercial degradation simulator for turbofan engines. It includes measurements that simulate failure under various operating conditions for several turbofan engines. A quick overview of the engine data sets FD001 and FD003 are shown in Table III as well as the respective fault conditions across multiple sensor measurements.

Each data set consists of 26 data columns. Columns 1 and 2 refer to the engine cycle for specific engines; Columns 3–5 denote the sensor measurements, such as temperature and pressure; the remaining columns reflect the simultaneous condition monitoring of 21 sensors. We apply standard data normalization [10] on the sensor data and assume equipment degradation behavior following (2). The objective of this experiment is to learn and consistently recommend an effective replacement action ϵ before imminent failure of turbofan engines, based on varying states of equipment health degradation.

VIII. RESULTS AND DISCUSSION

We shall briefly highlight the organization of results for reader's convenience, with details in the respective sections. First, we present the results from multiple experiments in order to highlight the benefits of PPO, based on MRS Game-1, and articulate the performance contributing factors for the policy network components in comparison to the baseline models and existing work. Second, we present results from the human participants and highlight important statistical findings. Table IV is then compiled to aggregate both the human participant and DRL results to illustrate the potential significance of augmenting human technicians in complex decision-making situations. In doing so, we demonstrate the applicability of PPO at the DAA level. Finally, we discuss the effectiveness of our proposed DRL system by presenting key results from C-MAPSS, which is utilized to mimic in situ decision-making at the equipment or ES layer. Furthermore, all model results reported in this article are the median average over five independent runs.

A. Performance Evaluation of Proposed Algorithm

Empirically, a deeper CNN design (i.e., PPO-CustomCNN) benefits from an increase of 7% in learning efficiency and 2% improvement in mean score deviations, when compared to the 3-layer CNN PPO algorithm (i.e., PPO-CNN). For baseline purposes, we hereby denote the nonclipped form

[2]https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#turbofan

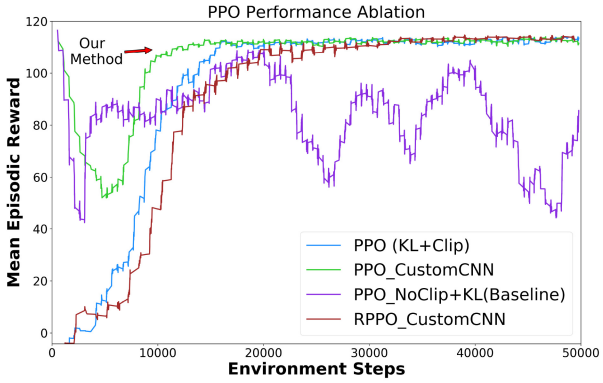

Fig. 7. Performance of our proposed PPO solutions.

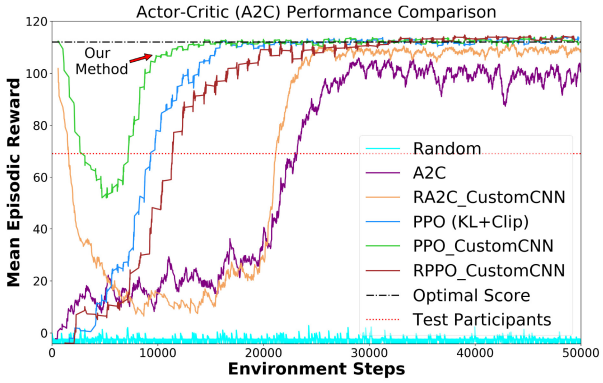

Fig. 8. Performance comparison between A2C variants, PPO variants, and human participants.

of PPO to be implicitly convergent (I.C.), and benchmark PPO's performance against the A2C variants, in Fig. 8. Notably, the proposed PPO variants are able to achieve up to 42% increase in learning efficiency. Besides, PPO reliably achieves an optimal score for Game-1, unlike the A2C variants which are merely near optimal. Besides, the performance of PPO with clipping is almost empirically identical to previous state-of-the-art results (i.e., A2C-LSTM). For completeness, we also include the PPO-CNNLSTM variant as part of our performance comparison, and the results are shown in Figs. 7 and 8.

Compared to atypical A2C networks, PPO is more robust to hyperparameter tuning. On the contrary, the PPO-CNNLSTM variant (i.e., RPPO_CustomCNN) requires a reasonably comprehensive hyperparameter tuning in order to achieve learning convergence. In other words, we discover that increasing the number of recurrent cells in the LSTM network to twice the step size of PPO yields the best results and convergence. One interesting insight from PPO-CNNLSTM variant is that it appears to tradeoff learning efficiency with higher overall scores as opposed to nonmemory-augmented PPO policies. A beneficial side effect of the LSTM network inclusion is the overall reduction of policy variance losses, and learning convergence variability in reward scores, as shown in Fig. 7. Across all four games, the PPO-CNNLSTM design on average outperforms both PPO-CNN and A2C-CNN-LSTM by 4% and 3%, respectively. For reader's convenience, we highlight

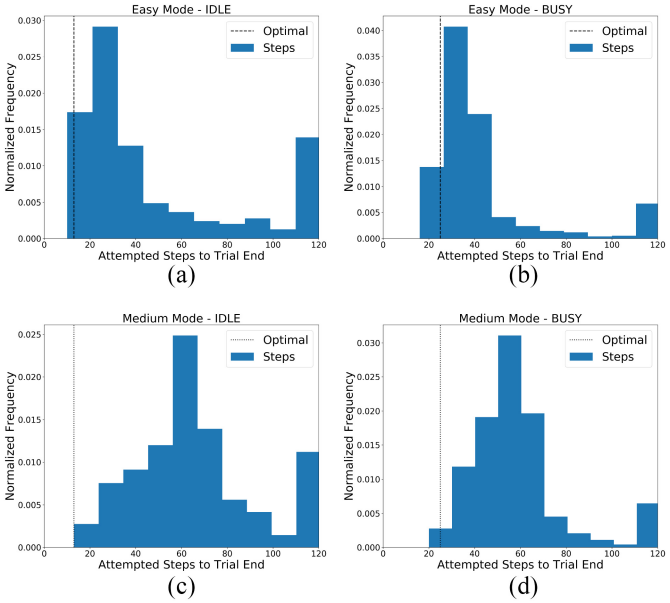

Fig. 9. Histogram analysis of human participant results (bins = 10). (a) Game 1. (b) Game 2. (c) Game 3. (d) Game 4.

the top performers per game, in bold, and the experimental results are listed in Table IV.

B. Human Participant Analysis

In relation to Fig. 8 and Table IV, the human participant scores are clearly suboptimal across all four games. By performing statistical analysis, we obtain insights into the human participant group's mean performance distribution as well as relative individual performance metrics in every game. For reference, the histogram analysis for all human participant performance, in all four games, is described in Fig. 9.

C. Equipment Severity Rating w.r.t. Technician Skill Level

With the encouraging results from Table IV, the PPO-CNNLSTM model evaluations are also conducted on the MRS simulator with all technician skill levels being effected across the range of equipment severity ratings. MTTR information can be derived from the rewards earned by each technician and, by normalizing the MTTR information, the probability of successful repairs can be obtained, see Fig. 10(a). Notably, these findings are reasonably consistent with our hypothesized risk-based state-transition relationship in Section IV-B.

The overall performance of PPO-CNNLSTM is empirically consistent between the senior and expert technicians, and its performance is characterized by considerations of the cost and availability of the manpower resources at any given time. For instance, once PPO-CNNLSTM identifies and dispatches a technician to repair the equipment, further resource substitution is disallowed, and the repair job must be completed by the dispatched technician, i.e., the DRL agent in this simulation case. Through iterative interaction with MRS, PPO-LSTM learns to dispatch the optimal skilled technician

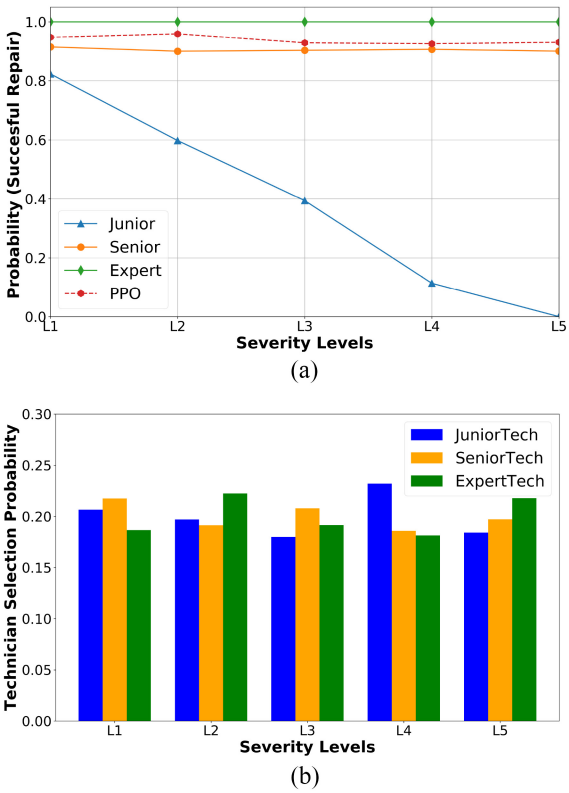

Fig. 10. Evaluating the decision-making effects of technician selection w.r.t. severity levels. (a) Repair probability per severity level. (b) Normalised PPO action probability.

so as to maximize its overall reward received at all severity ratings. Accordingly, it is suboptimal to select the expert level technician for the majority of severity ratings and its performance is justified by the severity level-based technician selection probabilities in Fig. 10(b).

To summarize the first two experiments, the potential benefits of DRL augmented human decision making for PdM action recommendation are clearly shown. Besides, PPO-CNNLSTM's improved learning efficiency results are due to the use of multiple actors, which utilize modern edge computing resources, such as multicore CPUs and GPUs, to realize a practical reduction in model training wall time when compared to similar DRL approaches.

D. C-MAPSS

When the original $N = 256$ learner hyperparameter is applied, poor performance convergence is observed because the multiple actors execute conflicting updates with our 256-step trace updates hyperparameter, causing PPO to behave like a Monte Carlo process [30]. Furthermore, by replacing the CNN module with two fully connected layer with 64 neurons each, the standalone PPO model achieves comparable performance to [10] with the added benefit of learning efficiency improvement of 73% (i.e., reduction from 12×10^3 to 3.2×10^3 time steps). Notably, the main hyperparameter for attaining learning convergence is to reduce the number of PPO actors to a single learner and environment.

TABLE IV
SAMPLE EFFICIENCY (LOWER IS BETTER) FOR A2C AND PPO ON FOUR GAME ENVIRONMENTS ARE SHOWN WITH AND WITHOUT THE PROPOSED OPTIMIZATIONS. PERFORMANCE RESULTS (HIGHER IS BETTER) OF EACH TEST WITH THE MEAN REWARDS OBTAINED FOR EACH BENCHMARK. FOR COMPARISON, THE HUMAN PARTICIPANTS SCORE ARE ALSO SHOWN

Game Modes	Time-steps to Learning Convergence (10^3)					Mean Reward Received					Human Participants Score
	A2C	PPO				A2C	PPO				
	CNN + LSTM	CNN	(w/CustomCNN)			CNN + LSTM	CNN	(w/CustomCNN)			
		w/ Clipping	PG No Clipping	Clipping	Clipping + LSTM		w/ Clipping	PG No Clipping	Clipping	Clipping + LSTM	
Game 1 (P(Fix)=1.0)	26	16	I.C.	**15**	32	108 ± 9.7	112 ± 6.2	88 ± 38.5	112 ± 6.1	**113 ± 5.5**	66 ± 47
Game 2 (P(Fix)=0.9)	27.5	**19**	I.C.	24	32	95 ± 9.5	98 ± 6.5	81 ± 28.8	**98 ± 6.2**	97 ± 7.1	72 ± 42
Game 3 (P(Fix)=0.6)	28	**12.5**	I.C.	15.5	21	143 ± 84.7	134 ± 55.1	92 ± 68.8	138 ± 46.8	**154 ± 48.3**	70 ± 52
Game 4 (P(Fix)=0.5)	31.5	**18.5**	I.C.	**18.5**	26	121 ± 63.4	115 ± 39	61 ± 57.9	114 ± 36.1	**118 ± 32.4**	88 ± 51

IX. CONCLUSION AND FUTURE WORK

In this article, we presented a DRL framework for an edge computing-based PdM model, to effectively manage the dynamic decision-making process involving equipment maintenance, maintenance cost model, and manpower resource. We formulated the complex resource management as a DRL problem for learning an optimal decision policy given a stochastic environment and time-series data. We evaluated the performance of the proposed PPO-LSTM using an MRS, and the findings are compared to those of human participants. The simulation results verify the efficacy of our framework and PPO-LSTM approach in addressing the challenging maintenance resource management problem, outperforming both human participants and the baselines in terms of convergence rate and performance. For future work, we plan to increase the learning efficiency of RL using knowledge transfer approaches, such as offline-to-online policy learning, continual learning, and transfer learning.

REFERENCES

[1] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," 2019. [Online]. Available: arXiv:1912.07383.

[2] M. Compare, P. Baraldi, and E. Zio, "Challenges to IoT-enabled predictive maintenance for industry 4.0," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4585–4597, May 2020.

[3] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[4] P. Wen, Y. Li, S. Chen, and S. Zhao, "Remaining useful life prediction of IIoT-enabled complex industrial systems with hybrid fusion of multiple information sources," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9045–9058, Jun. 2021.

[5] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, May 2017.

[6] P. Lin and J. Tao, "A novel bearing health indicator construction method based on ensemble stacked autoencoder," in *Proc. ICPHM*, San Francisco, CA, USA, 2019, pp. 1–9.

[7] C. Martinez, G. Perrin, E. Ramasso, and M. Rombaut, "A deep reinforcement learning approach for early classification of time series," in *Proc. 26th EUSIPCO*, Rome, Italy, 2018, pp. 2030–2034.

[8] Y. Ding *et al.*, "Intelligent fault diagnosis for rotating machinery using deep q-network based health state classification: A deep reinforcement learning approach," *Adv. Eng. Informat.*, vol. 42, Oct. 2019, Art. no. 100977.

[9] C. Zhang, C. Gupta, A. Farahat, K. Ristovski, and D. Ghosh, "Equipment health indicator learning using deep reinforcement learning," in *Proc. Joint ECML PKDD*, 2018, pp. 488–504.

[10] K. S. H. Ong, D. Niyato, and C. Yuen, "Predictive maintenance for edge-based sensor networks: A deep reinforcement learning approach," in *Proc. IEEE 6th WF-IoT*, New Orleans, LA, USA, 2020, pp. 1–6.

[11] J. Backman, J. Väre, K. Främling, M. Madhikermi, and O. Nykänen, "IoT-based interoperability framework for asset and fleet management," in *Proc. 21st Int. Conf. ETFA*, Berlin, Germany, 2016, pp. 1–4.

[12] Y. K. Teoh, S. S. Gill, and A. K. Parlikad, "IoT and fog computing based predictive maintenance model for effective asset management in industry 4.0 using machine learning," *IEEE Internet Things J.*, early access, Jan. 11, 2021, doi: 10.1109/JIOT.2021.3050441.

[13] K. S. H. Ong, W. Wang, T. Friedrichs, and D. Niyato, "Augmented human intelligence for decision making in maintenance risk taking tasks using reinforcement learning," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Oct. 2021.

[14] L. Li, Y. Peng, Y. Song, and D. Liu, "Lithium-ion battery remaining useful life prognostics using data-driven deep learning algorithm," in *Proc. PHM-Chongqing*, Chongqing, China, 2018, pp. 1094–1100.

[15] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications," *Mech. Syst. Signal Process.*, vol. 42, nos. 1–2, pp. 314–334, 2014.

[16] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention-based deep learning approach," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2521–2531, Mar. 2021.

[17] G. Zhao, G. Zhang, Y. Liu, B. Zhang, and C. Hu, "Lithium-ion battery remaining useful life prediction with deep belief network and relevance vector machine," in *Proc. ICPHM*, Dallas, TX, USA, 2017, pp. 7–13.

[18] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. ICPHM*, Denver, CO, USA, 2008, pp. 1–9.

[19] S. Bouajaja and N. Dridi, "A survey on human resource allocation problem and its applications," *Oper. Res.*, vol. 17, no. 2, pp. 339–369, 2017.

[20] G. Aydemir and B. Acar, "Anomaly monitoring improves remaining useful life estimation of industrial machinery," *J. Manuf. Syst.*, vol. 56, pp. 463–469, Jul. 2020.

[21] H. A. Dau *et al.*. (Oct. 2018). *The UCR Time Series Classification Archive*. [Online]. Available: https://bit.ly/3CjHcDj

[22] S. Panicucci *et al.*, "A cloud-to-edge approach to support predictive analytics in robotics industry," *Electronics*, vol. 9, no. 3, p. 492, 2020.

[23] S. Das, "Maintenance action recommendation using collaborative filtering," *Int. J. Health Policy Manage.*, vol. 4, no. 2, pp. 7–12, 2013.

[24] V. Katsouros, V. Papavassiliou, and C. Emmanouilidis, "A Bayesian approach for maintenance action recommendation," *Int. J. Prognost. Health Manage.*, vol. 4, p. 6, Oct. 2013.

[25] A. K. Farahat, C. Gupta, and H.-K. Tang, "System for maintenance recommendation based on maintenance effectiveness estimation," U.S. Patent 10 109 122, Oct. 23, 2018.

[26] A. Cachada *et al.*, "Maintenance 4.0: Intelligent and predictive maintenance system architecture," in *Proc. 23rd Int. Conf. ETFA*, vol. 1. Turin, Italy, 2018, pp. 139–146.

[27] J. Wang, L. Ye, R. X. Gao, C. Li, and L. Zhang, "Digital twin for rotating machinery fault diagnosis in smart manufacturing," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3920–3934, 2019.
[28] L. Decker, D. Leite, L. Giommi, and D. Bonacorsi, "Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Glasgow, U.K., 2020, pp. 1–8.
[29] B. Weber. *Predict The Unpredictable*. Accessed: Aug. 4, 2020. [Online]. Available: https://www.pwc.nl/nl/assets/documents/pwc-predictive-maintenance-4-0.pdf
[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[31] J. Beckmann and H. Heckhausen, "Situational determinants of behavior," in *Motivation and Action*. Berlin, Germany: Springer, 2018, pp. 113–162.
[32] P. Xiaolan, X. Lun, L. Xin, and W. Zhiliang, "Emotional state transition model based on stimulus and personality characteristics," *China Commun.*, vol. 10, no. 6, pp. 146–155, Jun. 2013.
[33] M. A. Thornton and D. I. Tamir, "Mental models accurately predict emotion transitions," *Proc. Nat. Acad. Sci.*, vol. 114, no. 23, pp. 5982–5987, 2017.
[34] J. E. Korteling, A.-M. Brouwer, and A. Toet, "A neural network framework for cognitive bias," *Front. Psychol.*, vol. 9, p. 1561, Sep. 2018.
[35] D. E. Berlyne, "The vicissitudes of aplopathematic and thelematoscopic pneumatology (or the hydrography of hedonism)," in *Pleasure, Reward, Preference: Their Nature, Determinants, and Role in Behavior*. New York, NY, USA: Academic, 1973, pp. 1–33.
[36] M. Sudhof, A. G. Emilsson, A. L. Maas, and C. Potts, "Sentiment expression conditioned by affective transitions and social forces," in *Proc. 20th ACM SIGKDD*, 2014, pp. 1136–1145.
[37] H. Xiang, P. Jiang, S. Xiao, F. Ren, and S. Kuroiwa, "A model of mental state transition network," *IEEJ Trans. Electron. Inf. Syst.*, vol. 127, no. 3, pp. 434–442, 2007.
[38] B. H. Prasetio, H. Tamura, and K. Tanno, "Deep time-delay Markov network for prediction and modeling the stress and emotions state transition," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 18071.
[39] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015. [Online]. Available: arXiv:1506.02438.
[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: arXiv:1707.06347.
[41] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. ICML*, 2015, pp. 1889–1897.
[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

Kevin Shen Hoong Ong (Member, IEEE) received the B.Eng. degree (Hons.) in electronics engineering with Management from the University of Dundee, Dundee, U.K., in 2007, and the M.Eng. degree from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2014, where he is currently pursuing the Ph.D. degree under the BOSCH-NTU Industrial Ph.D. program with the School of Computer Science and Engineering.

His research interests are in the areas of Internet of Things, deep reinforcement learning in predictive maintenance, resource allocation, and edge computing.

Wenbo Wang (Member, IEEE) received the B.S. and M.S. degrees from the School of Automation, Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in computing and information sciences from Rochester Institute of Technology, Rochester, NY, USA, in 2016.

He is currently a Research Fellow of the Faculty of Engineering, Bar Ilan University, Ramat Gan, Israel. Before that, he was with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include machine learning and mechanism design for multimedia wireless networks and Internet of Things.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkuts Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of Internet of Things, machine learning, and incentive mechanism design.

Thomas Friedrichs received the Ph.D. degree in nuclear physics from Technische Universität Braunschweig, Braunschweig, Germany, in collaboration with Institute Laue-Langevin–Grenoble, Grenoble, France, in 1998.

He is currently the Director of the IT Strategy and Innovation Asia Pacific, Robert Bosch (SEA) Pte Ltd., Singapore.