Intelligent Maintenance Framework for Reconfigurable Manufacturing with Deep Learning-based Prognostics

Tangbin Xia, *Member, IEEE,* Yimin Jiang, Yutong Ding, Guojin Si, Dong Wang, *Member, IEEE,*
Ershun Pan and Lifeng Xi

Abstract—**Future reconfigurable manufacturing systems (RMSs) can dynamically modify the system structures to achieve personalization, customization, and consumer-maker co-creation. The advanced Internet of Things (IoT) integrating deep learning and intelligent maintenance is critical for ensuring the operation and maintenance of RMS. On the one hand, a multi-head neural network is developed under the variability of individual machine degradations for deriving machine-level prognostics. It learns degradation features with superior generalization performance by simultaneously fitting multiple candidate distributions and updates remaining useful lifetime (RUL) distributions from diversified distribution ensembles. On the other hand, a flexible opportunistic maintenance policy is proposed to optimize the dynamic maintenance scheduling for the multi-phase RMS by utilizing real-time updated RUL distributions. Meanwhile, considering the unique attributes of RMSs, maintenance opportunities that arise from the in-situ machine PdM, system structure, and sequential reconstruction time are fully utilized. This IoT-enabled prognostics & opportunistic maintenance (POM) framework can achieve a bi-level interactive mechanism and inner/outer loops to reduce decision-making complexity and maintenance costs for future reconfigurable manufacturing. Numerical experiments demonstrate that this framework has superior predictive performance and significant cost savings to empower intelligent maintenance.**

Index Terms—**Deep learning-based prognostics; flexible opportunistic maintenance; lifetime distribution inferring; reconfigurable manufacturing system.**

I. Introduction

To accommodate personalized consumer demand and fierce market change, industrial production systems are evolving toward more reconfigurable, sophisticated, and flexible, which inevitably brings forward a higher request to ensure the availability and reliability of systems [1]. At the same time, industry and academia have focused on the reconfiguration concept to provide a large number of product requirements and rapid response to manufacturing adjustments. With the organic integration of Internet-of-Things (IoT) and sensor technologies, there has been increasing interest in the development of predictive maintenance (PdM) for facilitating cost-effective and uncompromised operations of complex manufacturing systems [2]. Accounting for the dynamic natures of engineering systems, PdM provides a practical, interactive framework for remaining useful lifetime (RUL) prediction and maintenance management [3]. Nevertheless, core characteristics of reconfigurable manufacturing paradigms, such as convertibility and scalability, broaden the variability of machines with varying degradation processes and performance parameters, impeding the optimal deployment of the current PdM [4]. On the other hand, these novel characters of reconfigurable structures bring new challenges for multi-phase maintenance scheduling and the speed of responsiveness, which is essential for operating machines in good condition. Hence, a reconfiguration-oriented prognostics & opportunistic maintenance (POM) framework is imperative that is promising to automatically process high-dimensional IoT data, adaptively infer prognostics information based on diverse structures, and empower system-level maintenance schemes for successive manufacturing phases.

For individual machines, one of the major tasks in PdM is predicting RUL to assess deterioration and provide the vital decision basis for subsequent maintenance plans. RUL is predicted based on a large amount of condition monitoring information collected through high-tech sensors embedded in IoT devices [5]. Unlike traditional intrusive sensors, infrared thermography (IRT) recently has been widely integrated into the sensing layer of the industrial IoT-enabled architecture for non-contact condition monitoring of functioning mechanical systems [6]. It can effectively estimate temperature distributions and capture thermographic patterns, which provide abundant spatiotemporal degradation information [7].

Based on infrared degradation image streams, AI-powered RUL prediction research can be grouped into two categories: conventional machine learning-based [8] and deep learning (DL)-based methodologies [9]. Lately, to release the reliance on manual labor in feature engineering, DL-based approaches such as recurrent neural networks (RNNs) and autoencoders, are employed to construct deep hierarchical architectures and automatically learn representative features [10]. Particularly, as the infrared image stream is recorded from a spatial domain over time, it describes a spatiotemporal degradation process

This work is supported by the National Key Research and Development Program of China under Grant 2022YFF0605700; the Natural Science Foundation of Shanghai under Grant 20ZR1428600; Oceanic Interdisciplinary Program of Shanghai Jiao Tong University under Grant SL2021MS008; and CSSC-SJTU Marine Equipment Forward Looking Innovation Foundation under Grant 22B010432. (Corresponding author: Guojin Si)

Tangbin Xia, Dong Wang and Lifeng Xi are with State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, SJTU-Fraunhofer Center, Shanghai 200240, China (e-mail: xtbxtb@sjtu.edu.cn; dongwang4-c@sjtu.edu.cn; lfxi@sjtu.edu.cn).

Yimin Jiang, Yutong Ding, Guojin Si and Ershun Pan are with Department of Industrial Engineering & Management, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: 415702168@sjtu.edu.cn; dingyutong_sjtu@sjtu.edu.cn; siguojin@sjtu.edu.cn; pes@sjtu.edu.cn).

that has complex characteristics including spatial propagation and non-separable covariance structures [11]. Notably, 3D convolutional neural networks (CNNs) exhibit impressive capabilities in simultaneously capturing the spatiotemporal information encoded in multiple adjacent infrared images [12]. Moreover, unlike the traditional single-output network, the multi-head prognostic network has become an effective technique to jointly predict diverse target attributes. Mo et al. [13] constructed a multi-head CNN-RNN prognostic model where each convolutional head had its specialized filters to learn appropriate representation. Xiang et al. [14] developed a multi-head gated recurrent unit with different learning modes to self-adaptively learn the degradation characteristics.

Despite the remarkable achievements of DL-based prediction techniques, most of the related work suffers from purely providing point estimations, which hamper dynamic optimization in subsequent systemic maintenance scheduling [15]. Prediction tasks are inherently subject to potential data uncertainty caused by measurement systems (such as sensor errors, noise contamination, and limited image resolution), which fail to provide sufficient insights into the real mechanical state [16]. Accordingly, RUL distributions of individual machines are preferable to measure prognostic uncertainties and quantify the risk of unexpected failures [17]. Bayesian NN (BNN) is promising to encompass uncertainty in DL-based prognostic applications, which place probabilistic priors upon the network parameters and infer their posterior distribution to approximate predictive uncertainties [18]. However, the quality of prognostic uncertainties measured using BNNs is affected by the intractability of posterior hypothesis reasoning and the computational effort of sampling at inference [19]. Therefore, to more conveniently deliver uncertainty qualification, deterministic NNs accompanied with additional components to predict the RUL distribution parameters of each machine were further developed [20].

For manufacturing systems, maintenance is one of the main costly items, where maintenance costs can typically account for between 15% and 70% of operating costs [21]. Specifically, maintenance expenditures may be wasted due to unnecessary, unplanned, or inappropriate maintenance activities, which requires PdM scheduling to be incorporated throughout the prognostics and health management in industrial IoT [22]. To stay competitive, increasingly companies possess reconfigurable manufacturing systems (RMSs) that are cost-effective and responsive to unpredictable and high-frequency market changes. In an advanced RMS, various types of machines with different reliability parameters and degradation processes constitute the reconfigurable structure. It is understood that reconfiguration includes not only adding/removing machines to/from the system but also replacing one machine with another. Accordingly, the rapid development of RMSs has forced decision-makers to propose accompanying dynamic maintenance policies to satisfy the maintenance requirement of the system with changeable structures. Xia et al. [23] developed an opportunistic maintenance policy for RMSs that comprehensively considered maintenance grouping and adjustment to respond rapidly to changeable system structures. Wang et al. [24] proposed a feasible integrated policy combining machine rearrangement and maintenance for a typical kind of reconfigurable system. Zhang et al. [25] addressed the dynamic modeling and control of the reconfigurable electronic assembly line under stochastic production disruption. Notably, these valuable policies are constructed using age-based population-specific reliability distributions obtained from historical data other than condition-based machine-specific information with updated prognostic information. For PdM, it is a philosophy or attitude that simply means using real-time prognostic information of machines and systems to optimize the overall operation of the manufacturing systems [26]. Therefore, a comprehensive IoT-enabled PdM framework uses advanced sensor technology to capture the operational status of critical systems and schedules all maintenance activities on demand based on this actual data.

To the best of our knowledge, several crucial challenges remain unsolved in the current PdM. Firstly, a comprehensive framework is unexplored considering DL-based prognostics, reconfiguration characteristics, and opportunistic maintenance. Moreover, probabilistic RULs are generally characterized by predetermined distributions such as the Gaussian [17] and the Weibull [19], which lack adequate self-adaptation and appropriate calibration. For one thing, since the weights of NNs are generally optimized by maximum likelihood estimation, predetermined RUL distributions inevitably bring tendentious influences to feature learning and could lead to model misspecification [18]. Meanwhile, the restricted options of RUL distributions are relatively knotty to handle the practical PdM demands of variable machines in RMSs [27]. Ignoring the heteroscedasticity and time-varying characteristic of the degradation processes of each machine results in a compromised distribution approximation, which cannot fully exploit IoT-enabled continuously monitored signals and increases the risk of unexpected failures. For another thing, only approximating the RUL distributions is insufficient for safety-critical PdM, where a poorly-calibrated interval estimation might not cover RULs with expected confidence intervals and occur either underconfident or overconfident prognostics [28]. To this end, further ensuring the authenticity of estimated prognostic distributions is essential.

To fill the mentioned research gaps, an IoT-enabled intelligent POM framework is developed that updates RUL distributions at the machine level and further effectively determines reconfiguration-oriented operation and maintenance schedules at the system level. The significant contributions are summarized as threefold:

1. Sufficiently considering degradation processes' stochastic nature and heteroscedasticity, a multi-head CNN (MHCNN) is developed to flexibly infer and dynamically update RUL mixture distributions rather than predetermination. Its backbone effectively learns generalized features and its private regressors synergetically capture the characteristics of different candidate distributions. Meanwhile, regarding degradation discrepancy in individual machines, mixture weights are continuously adjusted according to the input-dependent Akaike information criterion (AIC), making inferred distributions for each machine highly tailored.
2. A recalibration procedure is established to adjust the pre-

dicted RUL intervals reliably. This procedure continuously outputs a good approximation of the actual failure probability, which supplies trustable decision bases for the subsequent decision-making about the moment for performing operation and maintenance schemes.

3. Based on the real-time RUL updates, the configuration-driven flexible opportunistic maintenance policy is proposed for RMSs. Meanwhile, various maintenance opportunities and upstream/downstream relationship stability are comprehensively considered in system-level PdM separation and combination. The comparison results show that the IoT-enabled POM framework can help maintain the RMS in a timely and efficient manner.

The rest of this paper is structured as follows. Section II describes the decision-making processes of the IoT-enabled POM framework. Section III illustrates MHCNN that provides reliable RUL distributions. Section IV presents the flexible opportunistic maintenance policy. Experiments are investigated in Section V. Finally, Section VI concludes this paper and discusses potential future extensions.

II. IOT-ENABLED PROGNOSTICS & OPPORTUNISTIC MAINTENANCE FRAMEWORK

Advances in IRT sensor technology and wireless communication play a vital role in enabling IoT and monitoring the functioning of manufacturing processes. In this paper, we consider an RMS for multi-phase production, which includes different types of machines, where the number of machines and the system structure change periodically. Each machine is subjected to cumulative continuous degradation, causing stochastic failures. Machines in the RMS are interconnected in a multi-layer hierarchy. A stoppage of each machine due to anticipated PdM actions or unplanned failures leads to a shutdown or imbalance of the system. Further, different manufacturing phases correspond to different system structures (e.g., series, parallel, and series-parallel). Therefore, it is necessary to consider the impact of system structures and machine dependence on the maintenance decision-making and output the optimal maintenance schemes to ensure cost savings and production balance.

As shown in Fig. 1, this paper proposes an IoT-enabled POM framework encapsulating DL techniques, opportunistic maintenance, and dynamic decision-making in reconfigurable manufacturing scenarios. Specifically, MHCNN at the machine level dynamically predicts well-calibrated RUL distributions. After obtaining the real-time RUL distribution for each machine, a flexible opportunistic maintenance policy is developed to schedule the dynamic maintenance schemes of the RMS, considering the dependencies between machines. The general assumptions are considered as follows:

- At time $t = 0$, the original RMS design can be considered as the first reconfiguration, and the RMS starts operating with the initial system structure.
- The machines in the RMS are independent of individual machine degradations. If there is no PdM intervention, the degradation rate will increase as the machine ages. And the maintenance resources (technicians, tools, and so on) are always sufficient.

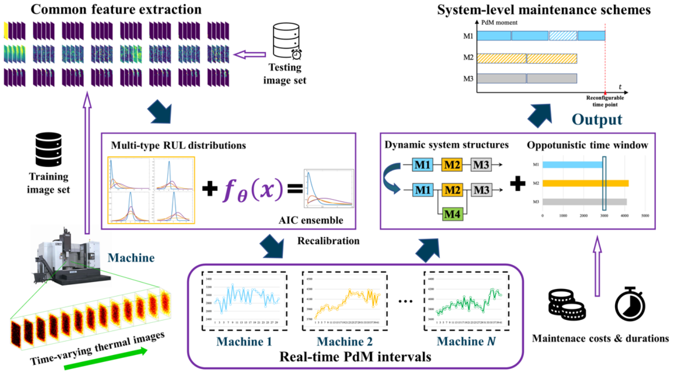

Common feature extraction
Testing image set
Training image set
Machine
Time-varying thermal images
Multi-type RUL distributions
$+ f_\theta(x) =$
AIC ensemble
Recalibration
Machine 1
Machine 2
Machine N
Real-time PdM intervals
Dynamic system structures
M1
M2
M3
M4
Oppotunistic time window
System-level maintenance schemes
PdM moment
t
Reconfigurable time point
Output
Maintenace costs & durations

Fig. 1. IoT-enabled POM process.

- According to the changing needs in terms of capacity and functionality, open-ended reconfigurations separate the production process into sequential manufacturing phases.

III. DEEP LEARNING-BASED PROGNOSTICS

Exploiting the IoT-enabled architecture, the overall objective of the DL-based prognostics approach aims to infer trustable RUL distributions of variable machines in RMS for supporting long-horizon operation and maintenance schedules. Firstly, MHCNN extends the RUL distribution inference to more data-dependent and flexible scenarios by simultaneously modeling and aggregating the nonlinear mappings between multiple candidate distributions and degradation signals, as described in Section III-A. Then, the recalibration procedure is introduced to modify the pre-trained MHCNN and solve the miscalibration of confidence intervals, as described in Section III-B. Eventually, real-time RUL distributions based on online monitored degradation signals of IoT-equipped machines can be inferred, updated, and further employed to obtain PdM intervals dynamically.

A. Multi-head CNN Integrating Uncertainty

As shown in Fig. 2, MHCNN is developed to reason RUL distributions and promote prognostic performance adaptively. It can emulate a broad spectrum of probability distributions of the nonnegative random variable by simultaneously fitting multiple candidate distributions and a diverse ensemble of distributions. Practitioners receive historical training samples $S = \{(x_n, r_n)\}_{n=1}^{N}$ from $X \times R \subseteq S$, where x_n is the input degradation signal sample, and $r_n \in R$ is the output space of RULs. Particularly, the time-to-failure range and the diversity in working conditions of collected training data should cover unseen testing data as far as possible combinedistorical experience. Besides, testing RUL exceeding the limits in the training data is approximated as the mximum training RUL due to the machines functioning normally at this moment [29].

With input samples, a backbone C with weight parameters θ_C consisting of stacked convolutional layers is constructed as a common feature extractor to learn features tailored to diversified distributions and raise the generalization performance. Second, depending on common features, multiple private regressors P_i with weight parameters θ_{P_i} are introduced

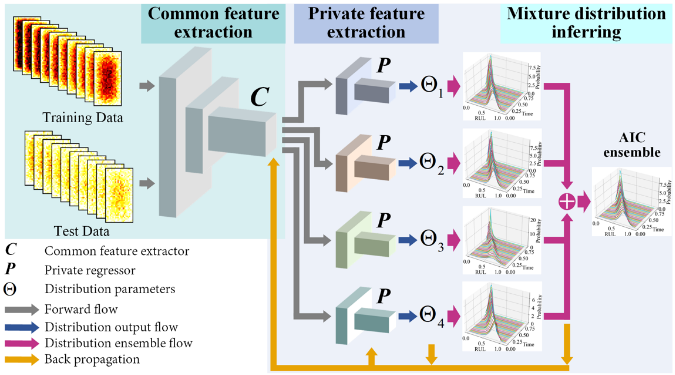


Fig. 2. The architecture of the MHCNN.

to define different probabilistic distributions for manipulating predictive uncertainties. Third, to endow substantial flexibility for distribution inference, a mixture distribution is adaptively constructed through heterogeneous ensembles employing the AIC to output comprehensive prognostic distribution estimations and uncertainty measurements. Considering errors between the predicted and actual RULs, and the negative log-likelihood loss with each parameterized distribution, MHCNN is continuously updated until it reaches a preset number of iterations.

With input samples, a backbone C with weight parameters θ_C consisting of stacked convolutional layers is constructed as a common feature extractor to learn features tailored to diversified distributions and raise the generalization performance. Second, depending on common features, multiple private regressors P_i with weight parameters θ_{P_i} are introduced

to define different probabilistic distributions for manipulating predictive uncertainties. Third, to endow substantial flexibility for distribution inference, a mixture distribution is adaptively constructed through heterogeneous ensembles employing the AIC to output comprehensive prognostic distribution estimations and uncertainty measurements. Considering errors between the predicted and actual RULs, and the negative log-likelihood loss with each parameterized distribution, MHCNN is continuously updated until it reaches a preset number of iterations.

In the context of distribution regression, the target r^* with a given sample $x^* \in X$ is assumed to be drawn independently and identically from a heteroscedastic RUL distribution. Generally, the probabilistic RUL predictive density $p(r^*|x^*)$ is not readily available and is promising to be approximatively represented conditioning on the sampled training dataset S. Each branch (including the backbone C and a private regressor P_i) of MHCNN acts as a nonlinear function f_{θ_i} for estimating $p(r^* \mid S, x^*)$, where $\theta_i = \{\theta_C, \theta_{P_i}\}$. Specifically, considering a maximum likelihood viewpoint, the network parameters are optimized by maximizing the likelihood of N paired training samples. Different from traditional NNs, which directly provide point estimations, the final layer of each branch infers distribution parameters $\theta_i = f_{\theta_i}(x)$ where the RUL distribution is parameterized as $p(r|\theta_i)$. MHCNN is scalable in predictive uncertainty estimation as each branch can accommodate a type of distribution. Meanwhile, despite arbitrary distributions being available to characterize $p(r|\theta_i)$, some candidate distributions prevalently leveraged across numerous applications in reliability engineering, such as the Gaussian, the Logistic, the Weibull, and the Smallest Extreme Value (SEV) distribution, are preferable for modeling machines' degradation. Consequently, the above four types of distributions are introduced (i.e., $i = 1, 2, 3, 4$), and the learning of the four branches is implemented through the minimization of the below negative log-likelihood loss functions:

$$L(\theta_1) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{(r_i-\mu_{1n})^2}{2\tau_{1n}^2}+\frac{1}{2}\log(2\pi\tau_{1n}^2)\right) \tag{1}$$

$$L(\theta_2) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{r_i-\mu_{2n}}{\tau_{2n}}+\log\tau_{2n} * \left(1+e^{-\frac{r_i-\mu_{2n}}{\tau_{2n}}}\right)^2\right) \tag{2}$$

$$\begin{aligned} L(\theta_3) = \frac{1}{N}\sum_{n=1}^{N}\Bigg(&-\log\frac{\tau_{3n}}{\mu_{3n}} \\ &+(1-\tau_{3n})\log\frac{r_n}{\mu_{3n}}+\left(\frac{r_n}{\mu_{3n}}\right)^{\tau_{3n}}\Bigg) \end{aligned} \tag{3}$$

$$L(\theta_4) == \frac{1}{N}\sum_{n=1}^{N}\left(\log\tau_{4n}-\frac{r_n-\mu_{4n}}{\tau_{4n}}+e^{\frac{r_n-\mu_{4n}}{\tau_{4n}}}\right) \tag{4}$$

where $\Theta_1 = \{\mu_1, \tau_1^2\}$ denotes the mean and variance parameters of Gaussian distribution. $\Theta_2 = \{\mu_2, \tau_2\}$ denotes the location and scale parameters of Logistic distribution. $\Theta_3 = \{\mu_3, \tau_3\}$ denotes the scale and shape parameters of Weibull distribution. $\Theta_4 = \{\mu_4, \tau_4\}$ denotes location and scale parameters of SEV distribution.

In optimizing θ_i, uncertainties in the stochastic nature of degradation processes can be preliminarily modeled by the above likelihood functions. However, two critical issues deserve further investigation. For one thing, maximizing likelihood over θ might learn features only tailored to a specific single type of predetermined distribution and be overfitting due to the deviation between the family of distributions being fit and the underlying data-generating process. For another thing, as shown in Fig. 2, the family of distributions individually predetermines the shape and statistical properties of predicted distributions. The Gaussian distribution is illustrated as a symmetric and unimodal curve. The Logistic distribution resembles the Gaussian distribution in appearance but has a longer tail and is more insensitive to atypical observations. The Weibull distribution is relatively conservative compared to the Gaussian distribution and suitable for machines with a high failure rate, whose density function is required to tend to zero at the observation time, increase until its mode, and decrease after it. Meanwhile, the SEV distribution is applicable to model the smallest value from a distribution whose tails decay exponentially quickly. As a result, in cases where a predetermined distribution is too restrictive to fit the distribution with heavy tails, higher kurtosis, and abnormal samples, a more flexible distribution (e.g., a mixture distribution) is worth considering.

In the proposed MHCNN, each branch is optimized as a mixture component learner to derive an adaptive mixture distribution. A collection of branches leads to heterogeneous ensembles for a convex combination of candidate distribution functions. Concerning the tradeoff between increasing the likelihood and avoiding overfitting, the mixture weight associated with each branch, φ_i, is determined by employing the AIC value and the softmax function, which measures the efficiency of a parameterized model as:

$$AIC_i = -2\log p(r \mid \widehat{\Theta}_l) + 2k_i \tag{5}$$

$$\varphi_i = e^{\frac{\frac{1}{AIC_i}}{\max(AIC)}} \Big/ \sum_{i=1}^{4} e^{\frac{\frac{1}{AIC_i}}{\max(AIC)}} \tag{6}$$

where k_i represents the number of parameters estimated in the regressors P_i. φ_i is calculated based on the AIC values and the softmax function, where divided by $\max(AIC)$ to avoid the computational overflow problem with the softmax function. Consequently, a branch with a smaller AIC value is assigned a larger weight, encouraging the model to enhance relatively appropriate ones during optimization and improve the comprehensive fitting performance. Meanwhile, the softmax function normalizes the collection of the AIC values to a probability distribution, resulting in the weights being nonnegative and summing to 1.

The final mixture variable is also continuous, as the distributions inferred by each branch are continuous. Given the multiple candidate distributions, the mixture distribution can be automatically estimated as $p(r \mid \Theta) = \sum_{i=1}^{4} \varphi_i p(r \mid \Theta_i)$, where $\Theta = \{\Theta_i\}_{i=1}^{4}$. The final prediction and variance based on $p(r \mid \Theta)$ can be represented as:

$$\mathbb{E}(r \mid \Theta) = \sum_{i=1}^{4} \varphi_i \int_{-\infty}^{\infty} r p(r \mid \Theta_i) \mathrm{d}r = \bar{r} \tag{7}$$

$$\mathrm{Var}(r \mid \Theta) = \sum_{i=1}^{4} \varphi_i \int_{-\infty}^{\infty} (r - \bar{r})^2 p(r \mid \Theta_i) \mathrm{d}r \tag{8}$$

The proposed mixture distribution can enhance flexibility in RUL distribution inference which is promising to exhibit non-trivial higher-order moments, including skewness, long tails, and multimodality, making up for the absence of these properties within the components themselves. Furthermore, to capture complicated cross-correlation relationships and enhance interactivity among candidate distributions, the mean square error (MSE) between final predictions and actual RUL values is minimized, which can be computed as:

$$L_{\mathrm{MSE}} = \sum_{n=1}^{N} (\bar{r}_n - r_n)^2 / N \tag{9}$$

Employing the chain rule and the above loss functions, the Adam algorithm is recommended to update and optimize each part of MHCNN. This joint optimization of multiple loss functions based on backpropagation considerably promotes the generalization performance of the backbone and enables each branch to fit data as much as possible. On top of this, as the machine approaches failures, the degradation trends become more significant, and the predicted distributions become more concentrated, which implicitly accords to the monotonically decreasing property between the degradation uncertainty and RULs. These characteristics are crucial to laying an adequate foundation for further decision-making and ensuring broad applicability in advanced RMSs.

B. Recalibration Procedure for the Predicted Distribution

Purely reasoning RUL distributions without assuring their reliability and calibration is insufficient for safety-critical PdM (e.g., an estimated RUL interval with a confidence level of 80% generally has difficulty exactly containing 80% of observations) [30]. Mainly, ineffective interval estimation is a restrictive factor for optimal solutions pursuit of maintenance scheme. A recalibration procedure is introduced to match the empirical and prognostic cumulative probability distributions (CDFs) to address this problem of model misspecification. Specifically, a predicted mixture distribution $p(r_n|\theta)$ is predicted at x_n, and empirical CLs are defined as:

$$\hat{\delta}(F_n(r)) = \sum_{n'=1}^{N} I\{r_{n'} \leq F_n^{-1}(\delta)\}/N \tag{10}$$

where δ th quantile $F_n^{-1}(\delta) = \inf\{r : \delta \leq F_n(r)\}$ and $F_n(r)$ is the CDF targeting r_n computed by $\int_{-\infty}^{r} p(r_n \mid \Theta)$. $I\{\cdot\}$ denotes the indicator function, either 1 or 0 , depending on whether the condition is true. $\widehat{\delta}(F_n(r))$ records where each RUL lies concerning the model's predicted distribution for x_n. The prognostics calibration property defines that when N goes to infinity, if $\widehat{\delta}$ tends to δ for $\forall\delta \in [0, 1]$, the network will be called calibrated [28].

To yield a well-calibrated MHCNN, an auxiliary regressor AR is trained to further modify prognostic CDFs in the recalibration procedure. It aims to ensure that 100* δ% of observations fall below the δ-quantile of the CDF that MHCNN predicted for each RUL. Consequently, the prognostic CDF $F_n(r_n)$ is recorded i.e., where each r_n falls in the CDF $F_n(r)$. Furthermore, the empirical CDFs across all RULs are calculated as:

$$\hat{P}(F_n(r_n)) = \sum_{n'=1}^{N} I\{F_{n'}(r_{n'}) \leq F_n(r_n)\}/N \tag{11}$$

where $\widehat{P}(F_n(r_n))$ aims to estimate the ground-truth probability of a random $\mathcal{R}$ lying in $(-\infty, F_{\mathcal{X}}^{-1}(\delta))$ from collected data which is monotonically increasing.

$\hat{P}(F_n(r_n))$ compares each RUL to all other RULs and counts the fraction of the RULs whose value of the CDF is less than or equal to that of the RUL of interest. Then, a recalibration dataset is constructed as $\{F_n(r_n), \widehat{P}(F_n(r_n))\}_{n=1}^{N}$. This dataset represents that while r_n lies at the $F_n(r_n)$ quantile for its individual CDF, but it lies at the $\hat{P}(F_n(r_n))$ quantile of the CDF across all RULs. Through the mapping between $F_n(r_n)$ and $\hat{P}(F_n(r_n))$, AR modifies each RUL's predicted CDFs to be consistent with the true CDFs inferred empirically from data. Isotonic regression is an advisable candidate for AR, which can fit a non-decreasing line as close as possible to the empirical CDFs in a non-parametric algorithm [28]. And

Bagging algorithm is integrated into AR to prevent isotonic regression. AR flexibly makes up for the discrepancy between prognostic and empirical CDFs by a free-form line rather than fitting a straight line onto the points by force. Thus, AR transforms the quantiles of the estimated interval to produce a well-calibrated interval based on the distribution across RULs rather than for each specific observation.

The prognostic framework of RUL distributions is implemented by integrating online degradation signals acquired from individual IoT-equipped machines and the historical training dataset. In the offline stage, concerning the variability of degradation paths of different machines, the historical degradation data of machines whose time-to-failures are smaller than the current inspection time are excluded from the training dataset. Then, the MHCNN and recalibrator are trained based on the truncated historical training dataset. In the online stage, MHCNN infers and updates the RUL distributions based on real-time degradation signals after each maintenance action. Furthermore, the recalibrator calibrates the cumulative failure probability as $AR(F_n(r))$ based on the inferred RUL distribution and faithfully assess the predictive uncertainty. Noticeably, the calibrated cumulative failure probability is real-time updated when new observations of the degradation signals are accessible and further incorporated into the system-level maintenance scheduling.

IV. CONFIGURATION-DRIVEN FLEXIBLE OPPORTUNISTIC MAINTENANCE POLICY

Maintaining the reliability of RMSs is not only crucial for improving system performance but also valuable since the failure of a machine may induce irrecoverable events and affect product quality. The reliability of RMSs can be maintained at an acceptable level through optimized and flexible maintenance scheduling. Consider an RMS for multi-phase production, the machines' RUL distribution is reasoned through the MHCNN described in Section III. To implement maintenance schemes from the system-level perspective, we investigate the configuration-driven flexible opportunistic maintenance policy for RMSs. The principles of the proposed maintenance policy include: (1) In each manufacturing phase, changes in inter-dependencies between machines due to reconfiguration (including additions, removals, and replacements) need to be analyzed. (2) Maintenance scheduling must be centered on system configuration expansion and follow system structure changes. And (3) Maintenance scheduling needs to synthetically consider constraints such as reconfiguration time points and system structure updates. The system structure changes of the RMS can be demonstrated through an example, as shown in Fig. 3.

Based on the consideration of the impact of different system structures, this policy schedules opportunistic maintenance decisions based on the time window arising from each PdM opportunity. Specifically, in each manufacturing phase, the cost rate model is used to obtain individual PdM decisions based on machines' updated RUL distributions, thus providing opportunities for the current RMS, as described in Section IV-A. Subsequently, opportunistic time windows are utilized at each manufacturing phase to combine PdM actions in series subsystems and separate PdM actions in parallel subsystems to avoid unnecessary imbalance. Also, the dynamic width of time windows provides scalability for a flexible system structure in successive manufacturing phases, as described in Section IV-B.

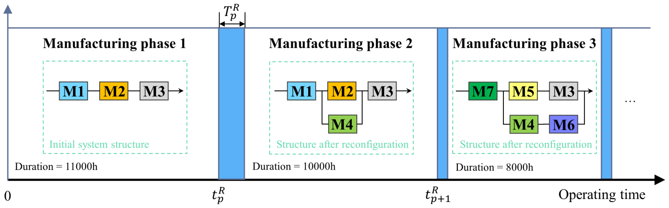


Fig. 3. An illustrative example of reconfiguration systems during successive manufacturing phases.

A. Individual PdM Decisions based on Prognostic Information

For each machine k, its degradation signals can be continuously monitored over time, which is used to provide a real-time RUL distribution. A comprehensive maintenance management framework needs to use the thermography tool to obtain the actual operating condition of machines and, based on this data, schedule all maintenance activities as needed. Including PdM in the maintenance management framework optimizes the availability of process machines and greatly reduces the cost of maintenance. Therefore, we use the expected maintenance cost rate function to optimize each machine's PdM interval, which is the period between two consecutive PdM actions. The function leverages machine-level RUL distribution to estimate the expected maintenance cost rate by considering the tradeoff between (1) ineffective use of machine lifetime by subjecting them to early maintenance and (2) increased risk of failure due to delayed maintenance. The function of the mth machine-level cycle for machine k is modeled as:

$$c_{km}(T_{km}) = \frac{C_{km}^P + C_{km}^F \cdot P(RUL \leq T_{km})}{T_{km} + T_{km}^P + T_{km}^F \cdot P(RUL \leq T_{km})} \tag{12}$$

where $c_{km}(T_{km})$ is the data-driven estimate for conducting maintenance activities during the PdM interval T_{km}. The cost factors C_{km}^P and C_{km}^F are the costs of planned PdM and failure replacement, respectively. Similarly, the time factors T_{km}^P and T_{km}^F are the time periods of planned PdM and failure replacement, respectively. And $P(RUL \leq T_{km})$ is the probability that machine k will fail in the current period. The individual PdM decision of each machine selects the time point corresponding to the lowest cost rate, denoted as T_{km}^*.

B. Maintenance Scheduling for Reconfiguration System

The system-level maintenance optimization scheduling for the reconfiguration system is established with the real-time pulling machine-level PdM intervals and dynamically changing system structures as inputs. Firstly, each decision phase is determined on the system operation timeline based on the reconfiguration time points t_p^R that appears through the sequence, which is the outer loop division of scheduling decisions. Based on the analysis of the newly reconfigured configuration, system structures are subdivided into sets of parallel and series subsystems, which is the update of the decision basis provided by the outer loop to the inner loop. After that, the maintenance time window is used to generate system-level maintenance scheduling schemes recursively in cycles based on the principle of PdM action separation and combination. Based on the results, PdM actions are implemented for the corresponding machines at each opportunity. Finally, a new round of system reconfiguration is triggered according to the new production changes, and the next manufacturing phase is started by returning to the outer cycle, and so on until the entire dynamically reconfigured system is terminated. The detailed steps are described as:

Step 1 - PdM interval pulling & system structure analysis: According to the optimal scheduling requirements of the system level, the machine level dynamically provides the optimal PdM interval of each machine in real-time and evaluates the values of the maintenance time window $V_p(\forall T^P_{km} < V_p < \forall T^*_{km})$. Carry out the dynamic maintenance scheduling from the first two-level cycle $m = 1$, $s = 1$. For the first system-level cycle, the PdM moments τ_{ks} are arranged based on the optimal PdM interval T^*_{km}.

$$\tau_{ks} = T^*_{km}, \quad m = 1, s = 1 \tag{13}$$

Step 2 - Inner PdM separation for parallel subsystems: During each manufacturing phase $M_p(p = 1, 2, 3, \dots)$, the updated V_p is utilized to separate the PdM in every $N^{parallel}$-machine parallel subsystem. The purpose is to refrain from simultaneous PdM actions of this parallel subsystem, which would lead to unnecessary upstream/downstream failures. Specifically, choose the first machine (index is marked as $k1$) that reaches its PdM moment as the system-level separation moment o_s.

$$o_s = \min(\tau_{ks}) = \tau_{(k1)s}, \quad 0 < k \leq N^{\text{parallel}} \tag{14}$$

Then, identify whether the arranged moment o_s is greater than or equal to the next reconfiguration time point t^R_{p+1}. If the answer is NO, which means $o_s < t^R_{p+1}$, check whether the time-window methodology combines the PdM action of all machines in this parallel subsystem by $\tau_{ks} \leq o_s + T^P_{(k1)m}(0 < k \leq N^{\text{parallel}}$, $k \neq k1)$. If the answer is YES, choose another machine $k2(k2 \neq k1)$ and separate the PdM action for machine $k2$ based on the time-window methodology to avoid downtime of the entire parallel subsystem due to maintenance. Therefore, the system-level PdM moment can be adjusted as:

$$\tau_{ks} = \tau_{(k2)s} = o_s + V_p \tag{15}$$

Meanwhile, if the answer about subsystem breakdown judgment is NO, execute the PdM action for machine $k1$, and reallocate the system-level PdM moment accordingly. End the current PdM separation and turn to Step 4 to process the system structure change for the following phase M_{p+1}.

$$\tau_{ks} = \tau_{(k1)(s-1)} + T^P_{(k1)m} + T^*_{km} \tag{16}$$

Step 3 - Inner PdM combination for series subsystems: Similarly, the updated V_p is also utilized to combine the PdM in every N^{series}-machine series subsystem. For a series subsystem, when one of the machines is subject to performing PdM, the shutdown of that machine means an interruption of production for all the machines in the subsystem. However, it also brings opportunities for other machines, and the simultaneous maintenance of multiple machines results in significant cost savings compared to the individual maintenance of each machine. Therefore, this step aims to maintain the machines within the time window into groups, and the specific implementation processes are as follows. Firstly, choose the first machine of this series subsystem that reaches its PdM moment as the system-level combination moment g_s, which can be described by:

$$g_s = \min(\tau_{ks}), \quad 0 < k \leq N^{\text{series}} \tag{17}$$

Then, identify whether the combination moment g_s is greater than or equal to the next reconfiguration time point t^R_{p+1}. If the answer is NO, which means $g_s < t^R_{p+1}$, distinguish between all machines to be maintained with the time window $[g_s, g_s + V_p]$ and machines that are not eligible for early maintenance. The corresponding discriminatory criteria for the PdM combination are shown in Eq. (20). PdM combination decision $x(k, g_s) = 1$ means the PdM action of machine k is combined to be conducted at a combination moment g_s, and 0 otherwise.

$$x(k, g_s) = \begin{cases} 0 & \tau_{ks} > g_s + V_p \\ 1 & g_s \leq \tau_{ks} \leq g_s + V_p \end{cases} \tag{18}$$

For the next $s + 1$ th system-level cycle, the PdM moments $\tau_{ks}(0 < k \leq N^{\text{series}}$) needs to be scheduled based on the PdM combination results. Assign $T^{P\max}_{(s-1)}$ as the maximum time for PdM actions combined in the s th cycle, which can be calculated by $\max\{x(k, g_{s-1}) \cdot T^P_{km}\}, 0 < k \leq N^{\text{series}}$.

$$\tau_{ks} = \begin{cases} \tau_{k(s-1)} + T^{P\max}_{(s-1)} & x\left(k, g_{s-1}\right) = 0 \\ g_{s-1} + T^{P\max}_{(s-1)} + T^*_{k(m+1)} & x\left(k, g_{s-1}\right) = 1 \end{cases} \tag{19}$$

Step 4 - Outer PdM combination for the whole system: At the next reconfiguration time point t^R_{p+1}, the machines will be adjusted for a new system structure within the duration T^R_{p+1} before stepping into the next manufacturing phase $M_(p + 1)$. Meanwhile, the sequential reconfiguration time point also creates opportunities for other non-repair machines. Considering the PdM combination for the whole system, the downtime within the manufacturing phase can be reduced. Thus, handhold this maintenance opportunity and judge the PdM actions that stay within the time window as follows:

$$y(k, t^R_{p+1}) = \begin{cases} 0 & \tau_{ks} > t^R_{p+1} + V_{p+1} \\ 1 & t^R_{p+1} \leq \tau_{ks} \leq t^R_{p+1} + V_{p+1} \end{cases} \tag{20}$$

Similarly, for the next $s + 1$th system-level cycle, the new PdM moments τ_{ks} of machines after outer PdM combination are updated as:

$$\tau_{ks} = \begin{cases} \tau_{k(s-1)} + T_{p+1}^{R} & y\left(k, t_{p+1}^{R}\right) = 0 \\ t_{p+1}^{R} + T_{p+1}^{R} + T_{k(m+1)}^{*} & y\left(k, t_{p+1}^{R}\right) = 1 \end{cases} \tag{21}$$

Step 5 - Cyclic maintenance scheduling for the next phase: The flexible structure reconfiguration will bring the next manufacturing phase M_{p+1}, while reconfiguring a brand-new structure includes adding new machines, removing old machines, and replacing machines with others. Before the system reaches its lifetime, return to step 2 to continue the PdM separation and combination for updated parallel/series subsystems for the new phase. For the newly added machine k', its PdM moment $\tau_{k's} = t_{p+1}^{R} + T_{p+1}^{R} + T_{k'1}^{*}$.

Step 6 - Total maintenance cost evaluation: According to the above opportunistic maintenance judgment using the time window, three different states exist for all machines, namely, (1) no PdM but is forced to shut down, (2) performing PdM action, and (3) no PdM and continuing operations. Decision variables $\phi_{ks} = 0, 1, 2$ denote three machine states accordingly. The proposed maintenance policy is to develop a scheduling process that can adapt to the dynamic changes in the system structure on the one hand and reduce maintenance costs while ensuring stable production through separation and combination analysis on the other hand. Therefore, the performance of the configuration-driven flexible opportunistic maintenance policy can be evaluated by its total maintenance cost. Specifically, the maintenance cost for machine k with $\phi_{ks} = 0$ is only caused by downtime for the PdM duration of other machines. The maintenance cost for machine k with $\phi_{ks} = 1$ includes PdM cost, expected failure replacement cost, and downtime cost. The maintenance cost for machine k with $\phi_{ks} = 2$ is zero. The expected maintenance cost for machine k at the sth system-level cycle is obtained by:

$$MC_{ks} = \begin{cases} D_k \cdot T_s^{P\max} & \phi_{ks} = 0 \\ \begin{aligned} C_{km}^{P} + C_{km}^{F} \cdot P(RUL \leq \tau_{ks}) \\ + D_k \cdot T_s^{P\max} \end{aligned} & \phi_{ks} = 1 \\ 0 & \phi_{ks} = 2 \end{cases} \tag{22}$$

In addition to the expected maintenance cost of each machine described above, the total maintenance cost of the reconfigurable system at its sequential manufacturing phases also includes the cost of downtime at each manufacturing changeover gap. The total maintenance cost for fixable reconfigurable systems is formulated as:

$$TMC = \sum_{k \in \mathcal{K}} \sum_{s \in \mathcal{S}} MC_{ks} + \sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}} D_k \cdot T_p^{R} \tag{23}$$

V. NUMERICAL EXPERIMENTS

This section provides a comprehensive experimental evaluation of the proposed IoT-enabled POM framework based on a simulated infrared image dataset. As shown in Fig. 3, an adaptive system involving seven individual machines goes through three manufacturing phases to satisfy changeable market demands with a variable system structure. Firstly, for the machine-level prognostics, the data generation mechanism and the assignment of TTF of each individual machine are introduced [11]. After that, the network architecture and optimization parameters of the MHCNN, and the prognostic performance metrics are demonstrated. Then, Section V-A investigates the proposed approach's prognostic results, including prognostic point estimations, uncertainty quantification, and calibration effectiveness. Finally, Section V-B shows the dynamic decision-making process with the opportunistic time window. Based on the maintenance parameters estimated by reliability engineers, the cost-effectiveness of the proposed maintenance policy is verified by comparing it with other maintenance policies. It presupposes that the underlying physical degradation of bearings used in machine spindles follows heat transfer processes [31]. In total, 400 run-to-failure processes are modeled. For process $j, \epsilon_j(H, W, T)$ denotes the pixel value at location (H, W) at frame T, where $H \in [0, 0.1]$ and $W \in [0, 0.05]$. The generation of $\epsilon_j(H, W, T)$ is governed by:

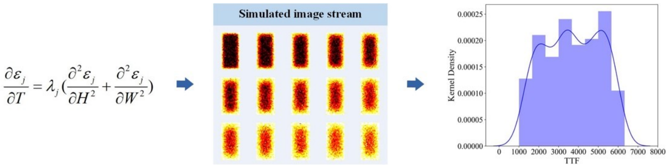


Fig. 4. Simulation details and the simulated image stream.

$$\frac{\partial \varepsilon_j}{\partial T} = \lambda_j \left(\frac{\partial^2 \varepsilon_j}{\partial H^2} + \frac{\partial^2 \varepsilon_j}{\partial W^2} \right) \tag{24}$$

where λ_j representing the diffusivity coefficient of machine j obeys a uniform distribution $\mathrm{U}(5 \times 10^{-6}, 1.5 \times 10^{-5})$. $\varepsilon|_{T=1} = 0$ and $\varepsilon|_{H=0} = \varepsilon|_{H=0.1} = \varepsilon|_{W=0} = \varepsilon|_{W=0.05} = 1$. The collection interval I is 100, and each stream sample contains ten consecutive images. The pixels and RULs are normalized into the range of $[0, 1]$, and the image shape is scaled to 4020, where these preprocessing operations would facilitate network optimization. The time-to-failure (TTF) for each process is assigned as

$$\mathrm{TTF}_j = (< \mathrm{vec}(\mathcal{B}), \mathrm{vec}(\varepsilon_j) > + \sigma * \epsilon_j) * I \tag{25}$$

where $\mathcal{B}$ stands for a set of coefficient tensors of rank 2. σ represents 5% of the standard deviation (std) of $< \mathrm{vec}(\mathcal{B}), \mathrm{vec}(\varepsilon_j) >$. ϵ_j follows the standard SEV distribution. Fig. 4 illustrates image generation details, including the TTF distribution of the above run-to-failure processes, which are multi-modal, broad, and unconcentrated. As a result, individual machines' degradation is according to heteroscedasticity and time-varying distributions.

As illustrated in Table I, the common feature extractor of MHCNN comprises 3D convolution layers (Conv3D), and each bi-head private regressor consists of fully connected (FC) layers. Conv3D and FC are normalized by the Batch normalization (BN) layer. Activations for the final layer of each regressor are selected based on the properties of its targeted RUL distribution, ensuring these predicted parameters are nonnegative. For fitting Gaussian, Logistic, and SEV distributions, location and scale parameters are respectively activated by the Sigmoid and Softplus functions. The Weibull distribution's predicted scale and shape parameters are activated by the Exponential and Softplus functions, respectively. Besides, the initial learning rate, the number of maximum epochs, and batch size are set as 0.01, 500, and 128 employing the grid search.

The model performance is evaluated from the prognostic accuracy and the uncertainty quantification. Besides MSE, the prognostic accuracy considers mean absolute error (MAE) and mean absolute percentage error (MAPE), which evaluate the absolute and relative error between the predicted and real RULs respectively. They are defined as:

$$L_{\mathrm{MAE}} = \sum_{n=1}^{N} \left|\bar{r}_n - r_n\right| / N, \tag{26}$$

$$L_{\mathrm{MAPE}} = \sum_{n=1, r_n \neq 0}^{N'} \left|\left(\bar{r}_n - r_n\right) / r_n\right| / N'. \tag{27}$$

The performance metrics for uncertainty quantification comprise prediction interval coverage probability (PICP), the mean prediction interval width (MPIW), their ratio, and the calibration error. PICP and MPIW assess the correctness and compactness of the estimated intervals, respectively. Their ratio is leveraged to evaluate the uncertainty quantification efficiency considering a broader MPIW would inevitably increase PICP. Particularly, the calibration error is introduced to assess the distance between expected CLs and the frequentist coverage probabilities, indicating the quality of the estimated intervals [28]. PICP, MPIW, and the calibration error are formulated as follows:

$$L_{\mathrm{PICP}} = \sum_{n=1}^{N} z_n / N, \tag{28}$$

$$L_{\mathrm{MPIW}} = \sum_{n=1}^{N} (C_u(x_n) - C_l(x_n)) / N, \tag{29}$$

$$\hat{P}(\delta_m) = \sum_{n=1}^{N} I\{F_n(r_n) \leq \delta_m\} / N, \tag{30}$$

$$L_{\text{Cal}} = \sum_{m=1}^{M} (\hat{P}(\delta_m) - \delta_m)^2, \tag{31}$$

where $z_n = 1$ if $C_l(x_n) \leq r_n \leq C_u(x_n)$ and 0 otherwise. $C_u(x_n)$ and $C_l(x_n)$ denote the upper and lower bound of the confidence interval of $x_n.M$ confidence levels (CLs), i.e., $0 \leq \{\delta_m\}_{m=1}^{M} \leq 1$ are partitioned with a step of 0.025 , and $\{\widehat{P}(\delta_m)\}_{m=1}^{M}$ are their corresponding empirical frequencies.

Meanwhile, prognostic experiments are performed based on Pytorch 1.2.0 using a computer with an NVIDIA GeForce RTX 3060 graphics card. 50% of run-to-failure processes are divided into the training subset, while the rest are partitioned into the testing subset. Each machine's health status is obtained by predicting the RUL of spindle bearings.

A. Machine-level Prognostic Technique Performance

Machine-level prognostic performance comparisons are conducted between the proposed approach and DL-based baseline models. Baseline models (D1 to D4) are implemented to respectively fit a predetermined single-type distribution, including the Gaussian, Logistic, SEV, and Weibull distributions. A baseline model (D5) is constructed to investigate the influence of quantifying uncertainty, which purely optimizes MSE. Each baseline model composes the same common feature extractor and a private regressor as MHCNN. The comparison with baseline models would demonstrate how increasing the flexibility of choosing RUL distribution types and the generalization performance of common features enhance the prognostic performance. Considering that Bayesian DL is also one of the most popular approaches for uncertainty quantification, three state-of-the-art BNNs are implemented for further comparison.

Firstly, we consider variational inference-based BNNs, which build an additional layer to define RUL distributions with a wider range, including the Gaussian (BDLG), Logistic (BDLL), and Weibull distribution (BDLW) [19]. In addtion, Monte Carlo dropout-based BNNs (including DPG1 [17] and DPG2 [18]) are also implemented, which achieve Bayesian approximation and model the uncertainty in terms of the Gaussian distribution. Besides Monte Carlo dropout, DPG1 considers stochastic characteristics of degradation processes and extra constructs an auxiliary network to quantify uncertainty jointly.

The selection of the BDL-based model aims to highlight the superiorities of MHCNN in terms of avoiding inaccurate approximate inference and relieving the computational effort of sampling. Besides, the uncalibrated and recalibrated models are compared regarding uncertainty quantification criteria and calibration performance to evaluate the recalibration method in promoting the reliability of estimated intervals.

1) Prognostic Accuracy Evaluation: The point estimation derives from the expected value of each predicted RUL distribution, one of the most representative statistics for evaluating prognostic performance. Table II and Fig. 5 qualitatively illustrate the mean and std of each model's prognostic accuracy. Particularly, D1 achieves the second-best accuracy among single-type distributions, which claims the reasonability of the traditional Gaussian distribution. Notably, no striking difference is noted among the baseline deterministic NNs until the proposed MHCNN breaks the performance bottleneck, which indicates the restrictiveness of the general types of RUL distributions. Additionally, the BNNs are 5.62% poorer concerning MSE on average than the single-type deterministic NNs, due to the inevitable growth in optimization difficulty and performance fluctuations brought by placing a probability distribution over model weights or applying too many dropout layers. BDLG and BDLL outperform other BNNs in terms of accuracy and stability, showing the necessity of appropriately predetermining the type of RUL distributions and introducing an auxiliary component to adapt to diversified distribution fitting demands. And BNNs requiring multiple sampling and extra dropout multiplications with the sampled mask during

TABLE I
CONFIGURATION PARAMETERS OF MHCNN.

Layer No.	Component type	Layer type	Kernel Size	Filter / Stride	Output size
1	Common feature extractor	Conv3D + BN + Dropout	(5, 5, 5)	32 / (2, 8, 4)	$[32, 5, 5, 5]$
2		Conv3D + BN	(3, 3, 3)	32 / (1, 1, 1)	$[32, 5, 5, 5]$
3		Conv3D + BN	(3, 3, 3)	32 / (1, 1, 1)	$[32, 5, 5, 5]$
4	Private regressor P_i	FC + BN + Dropout	(5)	–	[5]
5		Two FCs	(1)	–	[2]

TABLE II
THE RESULTS OF PROGNOSTIC ACCURACY.

		MHCNN	D1	D2	D3	D4	D5	BDLG	BDLL	BDLW	DPG1	DPG2
$L_{MSE}(\times 10^{-4})$	Mean	24.97	25.89	26.26	26.47	26.77	25.77	27.07	27.33	29.64	27.65	27.90
	Std	0.67	1.18	0.58	0.42	0.87	0.30	1.14	0.67	1.42	1.04	0.88
$L_{MAE}(\times 10^{-2})$	Mean	3.97	4.05	4.08	4.10	4.16	4.04	4.14	4.16	4.40	4.19	4.23
	Std	0.04	0.09	0.05	0.04	0.07	0.02	0.08	0.04	0.10	0.08	0.07
$L_{MAPE}(\%)$	Mean	45.95	48.67	46.09	48.38	61.60	48.83	52.32	51.90	74.22	61.75	63.71
	Std	2.59	4.60	5.49	4.48	6.08	2.31	2.86	3.09	2.90	5.68	3.90

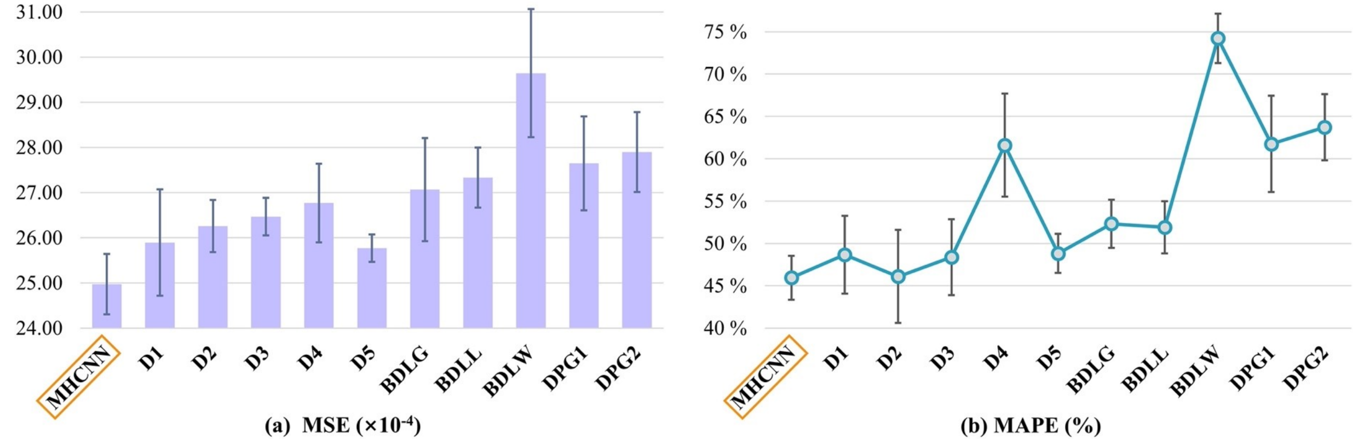
inference mean that MHCNN only needs a single forward pass per prediction, largely reducing computational expense.

Fig. 5 demonstrates that MHCNN stably obtains accurate outcomes compared with advanced BNNs and the baseline NNs, where the performance improvements come from two aspects. For one thing, single-type deterministic NNs converge towards one local optimum in the hypothesis space. BNNs further pay attention to the neighboring region of the optimum, where adjacent solutions around the optimum influence the prognostics on test samples. In contrast to these methods restricted to single regions, MHCNN, which segments a network architecture into varying sub-networks, approximates deep ensembles to realize multi-mode evaluation. Different private regressors in MHCNN are synergistically trained and integrated to derive a better generalization by converging to diverse local optima. A shared feature extractor is imposed to address various degradation processes' characteristics and increase the interactivity between multiple distributions while avoiding additional computation costs in feature extraction.

For another thing, for more general data-dependent and non-Gaussian scenarios, the actual RUL distribution usually is not in the candidate set, which requires a flexible model to reduce the risk of model misspecification in predetermination. MHCNN assigns mixture weights of candidate distributions according to AIC to fuse an appropriate RUL distribution for each sample. AIC is asymptotically optimal for choosing the distribution with the least MSE, facilitating the tradeoff between sub-networks with different goodness of fit. Table II shows that the mixture model improves predictability and is expressive in approximating distributions compared to a single predetermined distribution, alleviating the selection burden when the underlying RUL distribution is unknown.

2) Uncertainty Evaluation and Calibration Performance: Effective RUL distribution prediction is essential for practitioners to quantify prognostic uncertainty, estimate the cumulative failure probability, and develop optimal schedules to realize maintenance benefits. Table III summarizes the uncertainty evaluation results of the five most representative approaches with 95% CL, 90% CL, and 80% CL. It indicates that MHCNN provides a narrower MPIW while remaining a reliable PICP, resulting in the highest confidence interval efficiency. Fig. 6 visualizes the CL ratios for the five approaches. Although these Monte Carlo dropout-based approaches obtain the highest PICP, close to 100% with 95% CL, they seriously broaden the confidence interval, and their efficiencies are the lowest. Compared to MHCNN, the single-type deterministic D1 results in a broader confidence interval and lower efficiency due to the inability to capture non-Gaussian noises. It is observed that MHCNN is highly adaptive to changes in the degradation process, and it is competitive with popular Bayesian-based techniques in terms of these uncertainty quantification criteria. It also shows that Bayesian prior knowledge enhances uncertainty measurement capability, as BDLG outperforms dropout-based approaches. Moreover, as the ratio of DPG1 is more acceptable than DPG2, it proves that purely modeling the uncertainty inherent in random network parameters without the uncertainty from the degradation process is relatively limited for RUL prediction.

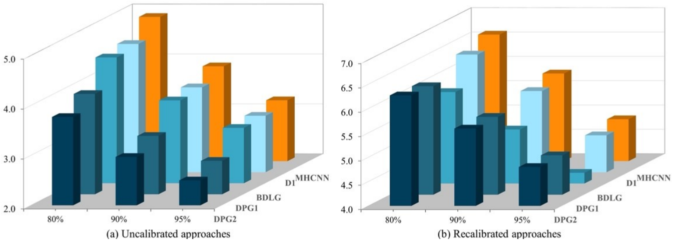
The prognostic results of one of the run-to-failure testing degradation processes visually illustrate the effect of constructing mixture distributions. Fig. 7 shows the predicted RUL distribution provided by different prognostic models. As time passes, the RUL distributions tend to shift toward zero, and their variances decrease. Baseline models obtain lower center densities than the Bayesian networks while appearing in similar shapes. Meanwhile, the distributions predicted by Monte Carlo dropout-based models have higher fluctuations. Notably, the distribution predicted by MHCNN during the initial phase of degradation is closer in resemblance to the Gaussian and SEV distributions, whereas, during the advanced phase of degradation, it is more akin to the Weibull and



(a) MSE ($\times 10^{-4}$)

(b) MAPE (%)

Fig. 5. MSE and MAPE results of MHCNN and other DL-based prognostic models.



(a) Uncalibrated approaches

(b) Recalibrated approaches

Fig. 6. Effectiveness ratio of the six models.

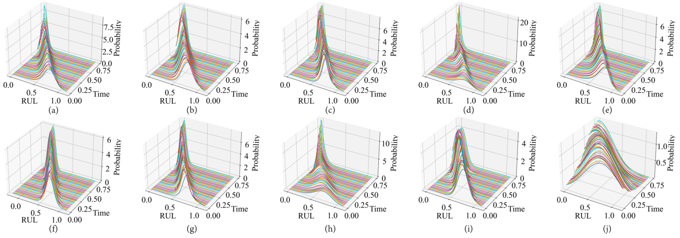


Fig. 7. Prognostic distributions. (a) MHCNN (b) D1 (c) D2 (d) D3 (e) D4 (f) BDLG (g) BDLL (h) BDLW (i) DGP1 (j) DGP2.

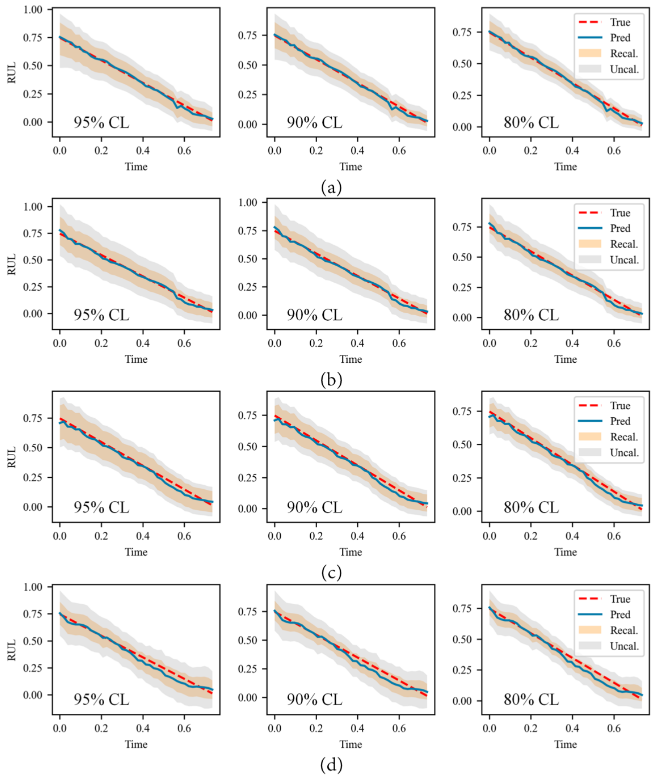
Logistic distributions. Thus, MHCNN takes full advantage of four base distributions, exhibiting better flexibility and fewer fluctuations in capturing various RUL distributions.

Table III illustrates that the mismatch between desired and predicted CLs commonly exists among RUL distribution prediction approaches. These approaches are prone to unnecessarily broadening confidence intervals for optimal PICP pursuit, decreasing efficiency, and inaccurate assessment of cumulative failure probability without appropriate constraints for interval estimation. It is demonstrated that the recalibration procedure shrinks the unreasonable confidence interval and gives reliable PICPs closer to the desired CLs, drastically increasing uncertainty quantification performance. The calibration error of recalibrated models is remarkably smaller than that of the original uncalibrated models. MHCNN and the conventional deterministic approaches are better calibrated than Bayesian DL-based approaches but are less authentic than the recalibrated ones.

Fig. 8 exhibits the deviation of the predicted RULs and the corresponding true values on the same run-to-failure processes with uncalibrated and recalibrated 80%, 90%, and 95% CLs. Firstly, it can be observed that all four RUL distribution prediction models are competent for capturing the degradation trend and providing an overall monotonic declining trend. Besides, it is especially noteworthy that predicted RULs of MHCNN are much closer to true RULs at the later stages of the degradation processes, which is vital for providing optimal maintenance guidance when machines approach their end of life. The uncalibrated MHCNN works significantly with better monotonicity and a narrower MPIW than traditional deterministic NNs and Bayesian-based approaches. Fig. 8 additionally shows a decreasing trend of the uncertainty of these four approaches as the machine degrades over time when the monitoring IoT sensor signals capture more evident degradation trends or patterns. This phenomenon stems from predicting a longer degradation trajectory more likely influenced by accumulated stochastic nature-induced noises. D1 and the dropout-based approaches deploy much wider confidence intervals and consistently overestimate the genuine uncertainty. In contrast, Bayesian DL approaches remain stable in point estimation but occasionally overestimate the true uncertainty. Particularly, well-calibrated MHCNN decreases the uncertainty value with improved prognostic accuracy and comprehensively maintains higher confidence interval efficiencies. After the recalibration procedure, all the confidence intervals shrink to correctly cover about 90% or 95% of the true RULs, maintaining a reliable balance between confidence interval compactness and correctness. Thus, with the joint optimization of MHCNN and the recalibration procedure, the machine-level prognostics empower a well-calibrated cumulative failure probability prediction for optimal solutions in pursuit of the system-level maintenance schedule.

B. System-level Flexible Maintenance Performance

With the updated RUL distributions from the machine-level MHCNN, dynamic maintenance scheduling based on various maintenance opportunities is conducted at the system level. Each individual PdM interval is arranged by a function of



(a)

(b)

(c)

(d)

Fig. 8. RUL interval prediction with varying CLs. (a) MHCNN (b) D1 (c) BDLG (d) DGP1.

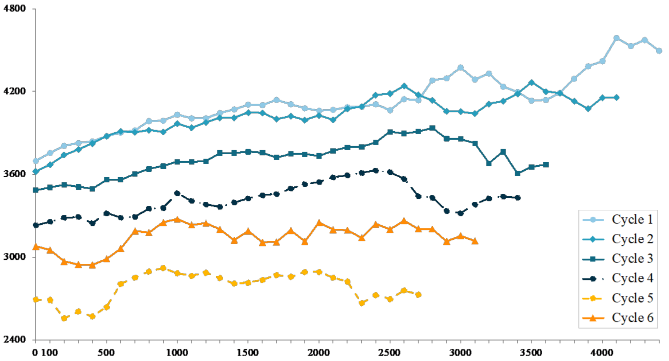


Fig. 9. Dynamic PdM intervals of machine 2 during the whole manufacturing process.

the expected maintenance cost rate shown in Eq. (12) to weigh ineffective use against the increased risk of failure. The detailed maintenance parameters are listed in Table IV. Based on the individual PdM decision process, the PdM interval of each machine is obtained once the RUL distribution is updated. The core motivation of PdM is to provide proven maintenance schemes for complex systems to ensure proper operation and reduce maintenance costs. Therefore, while the accuracy of machine-level prognostic techniques is important, it makes more sense from an economic perspective to translate the predictions into specific maintenance requirements. To better illustrate the real-time PdM requirement throughout the manufacturing process, Fig. 9 exhibits the dynamic PdM intervals of machine 2.

After obtaining the PdM interval, system-level maintenance scheduling continues to be optimized with various maintenance opportunities. Meanwhile, the width of the time window during each manufacturing phase is set to $V_1 = 200h$, $V_2 = 200h$, and $V_3 = 300h$. The reconfigurable system is opportunistically maintained according to the principle described in Section IV-B. After the dynamic maintenance scheduling for the current system-level cycle, we take note of the maintained and operational machines, update the rolling horizon of each machine, and reevaluate the degradation characteristics using the new degradation signal information. Sequentially, we reschedule the maintenance scheduling optimization for the next cycle and continue until the system reaches its lifetime. The duration between each manufacturing phase is set to $T_2^R = 500h$, $T_3^R = 200h$, and $T_4^R = 200h$. Accordingly, the system-level maintenance schemes for three-phase manufacturing are presented in Table V. Meanwhile, we define 5 types of markers to indicate the different determinations of machines:

- **SD** means the machine has no PdM action but is forced to shut down due to the maintenance of the other machine in the same series subsystem;
- **AM** represents advanced/postponed PdM moment based on the analysis of PdM separation or combination;
- **DN** indicates the machine is operational and doing nothing;
- **RM** presents the PdM action performed at the reconfiguration time point;
- **PdM** means the PdM action is performed based on the machine-level PdM interval.

Finally, to fairly evaluate the performance and cost-effectiveness of the proposed IoT-enabled POM framework, we benchmark it with five traditional maintenance policies using the same maintenance parameters and basic model. And the detailed maintenance principles are described as follows:

(1) ***Static individual maintenance policy (SIMP)***: Reliability characteristics from historical data are used for RUL prediction. The individual PdM intervals are used to arrange the maintenance plan without maintenance grouping. This benchmark assesses how far the IoT-enabled POM framework is improved by comprehensively considering the prognostics & opportunistic maintenance.

(2) ***Static opportunistic maintenance policy (SOMP)***: The RUL distribution is predicted only based on historical data. The system-level maintenance plan is optimized according to the criterion of opportunistic time window arising from the first PdM opportunity and the reconfigurable time point.

(3) ***Dynamic individual maintenance policy (DIMP)***: Pulling PdM intervals based on real-time RUL distribution. The PdM intervals are used to schedule the maintenance plan for each machine. The same as the SIMP framework, the state of the machine connected in series is SD, and the state of the machine connected in parallel is DN.

(4) ***Opportunistic maintenance policy with Monte Carlo dropout-based BNN (OMP-DPG2)***: The maintenance policy relies on Gaussian RUL distribution, obtained

TABLE III
UNCERTAINTY EVALUATION AND CALIBRATION PERFORMANCE OF THE FIVE APPROACHES

Model	CL	Uncalibrated model				Recalibrated model			
		L_{Cal}	L_{MPIW}	L_{PICP}	Ratio	L_{Cal}	L_{MPIW}	L_{PICP}	Ratio
MHCNN	95%	0.48%	30.89%	99.37%	3.22	0.03%	19.17%	93.09%	4.86
	90%		25.13%	97.87%	3.90		14.77%	85.72%	5.80
	80%		19.17%	93.65%	4.89		11.39%	75.19%	6.60
D1	95%	0.68%	32.01%	99.66%	3.13	0.04%	19.72%	93.83%	4.76
	90%		26.86%	98.71%	3.70		15.22%	86.29%	5.67
	80%		20.93%	95.20%	4.57		11.56%	74.16%	6.42
BDLG	95%	0.77%	32.09%	99.45%	3.11	0.38%	22.87%	96.44%	4.22
	90%		26.93%	98.29%	3.66		17.64%	90.01%	5.11
	80%		20.99%	94.70%	4.52		13.48%	79.25%	5.88
DPG1	95%	1.10%	37.46%	99.95%	2.67	0.05%	19.48%	93.42%	4.81
	90%		31.44%	99.70%	3.17		15.43%	86.22%	5.60
	80%		24.50%	98.08%	4.01		12.04%	74.89%	6.23
DPG2	95%	1.27%	40.01%	100.00%	2.50	0.08%	19.44%	93.16%	4.80
	90%		33.57%	99.78%	2.97		15.21%	84.94%	5.59
	80%		26.16%	98.49%	3.77		11.68%	73.20%	6.27

TABLE IV
MAINTENANCE PARAMETERS OF EACH INDIVIDUAL MACHINE

in Phase 1	in Phase 2	in Phase 3	$C^P_{km}(\$)$	$C^F_{km}(\$)$	$T^P_{km}(h)$	$T^F_{km}(h)$	$D_k(\$/h)$
Machine 1	Machine 1		3200	24000	500	120	35
Machine 2	Machine 2		5400	29000	360	100	40
Machine 3	Machine 3	Machine 3	4000	25000	800	180	75
	Machine 4	Machine 4	4400	27000	660	140	65
		Machine 5	6700	30000	700	240	80
		Machine 6	5800	24000	400	125	30
		Machine 7	4500	25000	570	150	50

TABLE V
SYSTEM-LEVEL MAINTENANCE SCHEMES DURING SUCCESSIVE MANUFACTURING PHASES.

	Opportunities (h)	M1	M2	M3	M4	M5	M6	M7
Phase 1	3217	PdM	SD	SD				
	4615	SD	PdM	AM				
	6317	PdM	SD	SD				
	8886	AM	AM	PdM				
	11000	RM	–	–				
Phase 2	13236	DN	PdM	DN	DN			
	13931	AM	SD	PdM	SD			
	15653	DN	DN	DN	PdM			
	16512	PdM	SD	SD	SD			
	17066	DN	PdM	DN	DN			
	18136	AM	SD	PdM	SD			
	19951	DN	AM	DN	PdM			
	20239	PdM	AM	SD	SD			
	21500	–	–	–	–			
Phase 3	22317			PdM	DN	DN	DN	DN
	24149			DN	PdM	AM	DN	DN
	24519			DN	DN	PdM	DN	DN
	25855			AM	SD	SD	AM	PdM
	27363			DN	DN	PdM	DN	DN
	28278			DN	PdM	DN	DN	DN
	29364			PdM	DN	DN	DN	DN

through a Bayesian deep-learning-based health prognostic approach [15]. Furthermore, the opportunities that arise from in-situ PdM and system reconfiguration are utilized to achieve maintenance optimization.

(5) ***Opportunistic maintenance policy with Monte Carlo dropout-based BNN (OMP-DPG1)***: Similarly, another prognostic approach [17], achieves the interval estimation of RUL and is used to generate the RUL distribution of each machine. Also, the two abovementioned benchmarks show the value of fitting multiple candidate distributions in effectively evaluating the dynamic RUL distribution.

Fig. 10 exhibits the column charts of the total maintenance cost achieved by the POM framework and the five benchmarks, namely: SIMP, SOMP, DIMP, OMP-DPG2, and OMP-DPG1. As shown in Fig. 10, the proposed POM is the least costly, while OMP-DPG1 and OMP-DPG2 come far second and third with a relative cost difference of 3.77% and 19.84%, respectively. The system-level maintenance scheduling demonstrates that the accuracy of machine RUL prediction directly affects the execution of system-level maintenance schemes. In our proposed POM, compared with existing prediction approaches, such as DPG1 and DPG2, MHCNN can help decision-makers get as close as possible to a better maintenance scheme.

Second, POM reduces upon DIMP by 18.01%. Another interesting observation is that with the same prediction approach (MHCNN), POM further reduces \$114703 by comprehensively considering the system structure and various maintenance opportunities. Meanwhile, the downtime cost is significantly reduced due to group maintenance. Hence, this trend also confirms the importance of group maintenance under various maintenance opportunities in multi-phase complex manufacturing systems. Third, DIMP, OMP-DPG2, OMP-DPG1, and POM significantly outperform "non-updating" frameworks like SIMP and SOMP, proving the importance of real-time machine RUL prediction.

The advantages of the IoT-enabled POM framework can be summarized as follows: Firstly, the real-time RUL prediction approach optimizes the number of PdM actions and unexpected failures because it uses updated degradation signal information to forecast RUL. Secondly, the POM framework accurately identifies which machine is triggering and advances other PdM actions according to the opportunistic time window. By integrating individual machine degradation and dynamic maintenance scheduling and designing a bi-level interactive mechanism and inner/outer loops, the IoT-enabled POM framework helps maintain changeable reconfigurable systems in a timely and efficient manner.

VI. CONCLUSION

This paper presents an IoT-enabled POM framework for reliable RUL distribution inference and configurable-driven dynamic maintenance scheduling. A DL-based prognostics approach is developed for RUL distribution inference, based on heterogeneous ensembles. Moreover, a flexible opportunistic maintenance policy is proposed to connect real-time RUL predictions and systemic PdM management. The characteristics of each RMS and various maintenance opportunities are comprehensively considered to arrange system-level maintenance schemes.

Meanwhile, its performance on simulated data indicates the proposed IRT-based prognostics and maintenance approach can be effectively employed for individual machines under different TTF distributions, system structures, and production processes with a good generalization. The machine-level experimental analysis demonstrates that MHCNN obtains superior prognostic accuracy, uncertainty measurement, and calibration performance. Furthermore, compared with five traditional maintenance policies, the system-level maintenance policy with real-time RUL distributions and opportunistic time windows responds rapidly to changeable system structures, significantly avoids upstream/downstream imbalance, and reduces downtime. When extending POM to other types of thermal data or multi-phase RMSs, several hyperparameters (including the convolutional kernel size of the backbone and the number of MHCNN heads) would be adaptively adjusted to fit changes in data.

In future work, potential topics will further optimize the data quality by separating anomaly data and dynamically updating machine-level prognostic models according to real-time monitoring data. In addition, opportunistic predictive maintenance should be expanded from a single maintenance activity to multiple maintenance activities.

REFERENCES

[1] M. Compare, P. Baraldi, and E. Zio, "Challenges to iot-enabled predictive maintenance for industry 4.0," *IEEE Internet of Things J.*, vol. 7, no. 5, pp. 4585–4597, 2019.

[2] 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 of Things J.*, vol. 10, no. 3, pp. 2087–2094, 2023.

[3] T. Xia, Y. Dong, L. Xiao, S. Du, E. Pan, and L. Xi, "Recent advances in prognostics and health management for advanced manufacturing paradigms," *Reliab. Eng. Syst. Saf.*, vol. 178, pp. 255–268, 2018.

[4] H. Tang, D. Li, J. Wan, M. Imran, and M. Shoaib, "A reconfigurable method for intelligent manufacturing based on industrial cloud and edge intelligence," *IEEE Internet of Things J.*, vol. 7, no. 5, pp. 4248–4259, 2019.

[5] Y. Jiang, T. Xia, D. Wang, X. Fang, and L. Xi, "Spatiotemporal denoising wavelet network for infrared thermography-based machine prognostics integrating ensemble uncertainty," *Mech. Syst. Signal Proc.*, vol. 173, p. 109014, 2022.

[6] R. A. Osornio-Rios, J. A. Antonino-Daviu, and R. de Jesus Romero-Troncoso, "Recent industrial applications of infrared thermography: A review," *IEEE Trans. Ind. Inform.*, vol. 15, no. 2, pp. 615–625, 2018.

[7] X. Li, H. Shao, S. Lu, J. Xiang, and B. Cai, "Highly efficient fault diagnosis of rotating machinery under time-varying speeds using lsismm and small infrared thermal images," *IEEE Trans. Syst. Man Cybern. - Syst.*, vol. 52, no. 12, pp. 7328–7340, 2022.

[8] Y. Song, S. Gao, Y. Li, L. Jia, Q. Li, and F. Pang, "Distributed attention-based temporal convolutional network for remaining useful life prediction," *IEEE Internet of Things J.*, vol. 8, no. 12, pp. 9594–9602, 2020.

[9] G. Aydemir and K. Paynabar, "Image-based prognostics using deep learning approach," *IEEE Trans. Ind. Inform.*, vol. 16, no. 9, pp. 5956–5964, 2020.

[10] X. Fang, K. Paynabar, and N. Gebraeel, "Image-based prognostics using penalized tensor regression," *Technometrics*, vol. 3, no. 3, pp. 369–384, 2019.

[11] X. Liu, K. Yeo, and J. Kalagnanam, "A statistical modeling approach for spatio-temporal degradation data," *J. Qual. Technol.*, vol. 50, no. 2, pp. 166 – 182, 2018.

[12] Y. D. A, T. X. A. B, D. W. A, X. F. C, and L. X. A. B, "Infrared image stream based regressors for contactless machine prognostics," *Mech. Syst. Signal Proc.*, vol. 154, no. 107592, 2021.

[13] H. Mo, L. L. Custode, and G. Iacca, “Evolutionary neural architecture search for remaining useful life prediction,” *Appl. Soft. Comput.*, vol. 108, no. 3, p. 107474, 2021.

[14] S. Xiang, Y. Qin, J. Luo, F. Wu, and K. Gryllias, “A concise self-adapting deep learning network for machine remaining useful life prediction,” *Mech. Syst. Signal Proc.*, vol. 191, p. 110187, 2023.

[15] K. T. Nguyen and K. Medjaher, “A new dynamic predictive maintenance framework using deep learning for failure prognostics,” *Reliab. Eng. Syst. Saf.*, vol. 188, pp. 251–262, 2019.

[16] E. Zio, “Prognostics and health management (phm): Where are we and where do we (need to) go in theory and practice,” *Reliab. Eng. Syst. Saf.*, vol. 218, p. 108119, 2022.

[17] M. Kim and K. Liu, “A bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics,” *IISE Trans.*, vol. 53, no. 3, pp. 326–340, 2020.

[18] W. Peng, Z.-S. Ye, and N. Chen, “Bayesian deep-learning-based health prognostics toward prognostics uncertainty,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2283–2293, 2020.

[19] G. Li, L. Yang, C.-G. Lee, X. Wang, and M. Rong, “A bayesian deep learning rul framework integrating epistemic and aleatoric uncertainties,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8829–8841, 2020.

[20] R. He, Z. Tian, and M. J. Zuo, “A semi-supervised gan method for rul prediction using failure and suspension histories,” *Mech. Syst. Signal Proc.*, vol. 168, p. 108657, 2022.

[21] M. Bevilacqua and M. Braglia, “The analytic hierarchy process applied to maintenance strategy selection,” *Reliab. Eng. Syst. Saf.*, vol. 70, no. 1, pp. 71–83, 2000.

[22] K. S. H. Ong, W. Wang, D. Niyato, and T. Friedrichs, “Deep-reinforcement-learning-based predictive maintenance model for effective resource management in industrial iot,” *IEEE Internet of Things J.*, vol. 9, no. 7, pp. 5173–5188, 2021.

[23] T. Xia, L. Xi, E. Pan, and J. Ni, “Reconfiguration-oriented opportunistic maintenance policy for reconfigurable manufacturing systems,” *Reliab. Eng. Syst. Saf.*, vol. 166, pp. 87–98, 2017.

[24] D. Wang, S. Si, Z. Cai, and J. Zhao, “Reliability optimization of linear consecutive-k-out-of-n: F systems driven by reconfigurable importance,” *Reliab. Eng. Syst. Saf.*, vol. 216, p. 107994, 2021.

[25] D. Zhang, M. Xie, H. Yan, and Q. Liu, “Resilience dynamics modeling and control for a reconfigurable electronic assembly line under spatio-temporal disruptions,” *J. Manuf. Syst.*, vol. 60, pp. 852–863, 2021.

[26] Y. Hu, X. Miao, Y. Si, E. Pan, and E. Zio, “Prognostics and health management: A review from the perspectives of design, development and decision,” *Reliab. Eng. Syst. Saf.*, vol. 217, p. 108063, 2022.

[27] S. Karimi, H. Liao, and N. Fan, “Flexible methods for reliability estimation using aggregate failure-time data,” *IISE Trans.*, vol. 53, no. 1, pp. 101–115, 2021.

[28] V. Kuleshov, N. Fenner, and S. Ermon, “Accurate uncertainties for deep learning using calibrated regression,” *Int. Conf. Mach. Learn.*, pp. 2796–2804, 2018.

[29] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, “Machinery health prognostics: A systematic review from data acquisition to rul prediction,” *Mech. Syst. Signal Proc.*, vol. 104, pp. 799–834, 2018.

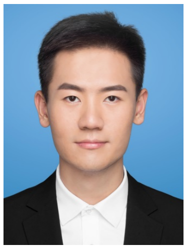
[30] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, p. 6405–6416, 2017.

[31] H. Yan, K. Paynabar, and J. Shi, “Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition,” *Technometrics*, vol. 60, no. 2, pp. 181–197, 2018.

VII. Biography Section

Tangbin Xia received Ph.D. degree in Mechanical Engineering (Industrial Engineering) from Shanghai Jiao Tong University in 2014. He worked as a postdoctoral in the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology and a joint Ph.D. student in the S. M. Wu Manufacturing Research Centre at the University of Michigan. He is currently an Associate Professor and serves as Deputy Director of the Department of Industrial Engineering at Shanghai Jiao Tong University. His major research interests are intelligent maintenance systems, prognostics & health management, and advanced manufacturing. He is a member of IISE, IEEE, ASME, and INFORMS.

Yimin Jiang received the B.S. degree in mechanical engineering from Xi’an Jiao Tong University, Xi’an, China, in 2019. He is working toward the Ph.D. degree with the State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include prognostic and health management of industrial equipment.



Yutong Ding received his B.S. degree in Central South University, Changsha, China in 2020. He is currently pursuing the Ph.D. degree in Mechanical Engineering (Industrial Engineering) with State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University. His current research interests include maintenance decision-making and condition-based maintenance.

Guojin Si received the Ph.D. degree in Mechanical Engineering (Industrial Engineering) from Shanghai Jiao Tong University in 2023. She is currently a Postdoctoral Fellow with the School of Mechanical Engineering, Shanghai Jiao Tong University. Her research interests include maintenance scheduling optimization for distributed systems and multi-period maintenance network design.

Dong Wang received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2015. He is currently an Associate Professor with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai. His research interests include sparse and complex measures, signal processing, prognostics and health management, condition monitoring and fault diagnosis, statistical learning, and nondestructive testing. Dr. Wang is an Editorial Board Member of Mechanical Systems and Signal Processing. He is an Associate Editor of the IEEE Transactions on Instrumentation and Measurement.

Ershun Pan received Ph.D. degree in Mechanical Engineering from Shanghai Jiao Tong University in 2000. He is currently a Professor and serves as Director of the Department of Industrial Engineering at Shanghai Jiao Tong University. He previously worked in the University of Michigan as a Visiting Scholar. His research and teaching interests are in the areas of the theory and methods of quality control, reliability engineering and maintenance strategy, and lean manufacturing technology.

Lifeng Xi received Ph.D. degree in Mechanical Engineering from Shanghai Jiao Tong University in 1995. He is currently a Professor and serves as Vice President of Shanghai Jiao Tong University. He also serves as the Editorial Director of Industrial Engineering & Management, and the Executive Director of the Chinese Quality Association. He has published more than 200 papers in international journals. His interest areas are in the fields of production system design and planning, quality management, and reliability engineering.