

Article

Data-Driven Predictive Maintenance Policy Based on Dynamic Probability Distribution Prediction of Remaining Useful Life

Shulian Xie ^{*}, Feng Xue , Weimin Zhang and Jiawei Zhu

School of Mechanical Engineering, Tongji University, No. 4800, Cao'an Highway, Shanghai 201804, China; xuefeng@tongji.edu.cn (F.X.); iamt@tongji.edu.cn (W.Z.); zhujiawei@tongji.edu.cn (J.Z.)

* Correspondence: shulian00@tongji.edu.cn; Tel.: +86-182-0198-9760

Abstract: As the reliability, availability, maintainability, and safety of industrial equipment have become crucial in the context of intelligent manufacturing, there are increasing expectations and requirements for maintenance policies. Compared with traditional methods, data-driven Predictive Maintenance (PdM), a superior approach to equipment and system maintenance, has been paid considerable attention by scholars in this field due to its high applicability and accuracy with a highly reliable quantization basis provided by big data. However, current data-driven methods typically provide only point estimates of the state rather than quantification of uncertainty, impeding effective maintenance decision-making. In addition, few studies have conducted further research on maintenance decision-making based on state predictions to achieve the full functionality of PdM. A PdM policy is proposed in this work to obtain the continuous probability distribution of system states dynamically and make maintenance decisions. The policy utilizes the Long Short-Term Memory (LSTM) network and Kernel Density Estimation with a Single Globally-optimized Bandwidth (KDE-SGB) method to dynamic predicting of the continuous probability distribution of the Remaining Useful Life (RUL). A comprehensive optimization target is introduced to establish the maintenance decision-making approach acquiring recommended maintenance time. Finally, the proposed policy is validated through a bearing case study, indicating that it allows for obtaining the continuous probability distribution of RUL centralized over a range of ± 10 sampling cycles. In comparison to the other two policies, it could reduce the maintenance costs by 24.49~70.02%, raise the availability by 0.46~1.90%, heighten the reliability by 0.00~27.50%, and promote more stable performance with various maintenance cost and duration. The policy has offered a new approach without priori hypotheses for RUL prediction and its uncertainty quantification and provided a reference for constructing a complete PdM policy integrating RUL prediction with maintenance decision-making.

Keywords: predictive maintenance policy; long short-term memory network; remaining useful life; data-driven; kernel density estimation; continuous probability distribution



Citation: Xie, S.; Xue, F.; Zhang, W.; Zhu, J. Data-Driven Predictive Maintenance Policy Based on Dynamic Probability Distribution Prediction of Remaining Useful Life. *Machines* **2023**, *11*, 923. <https://doi.org/10.3390/machines11100923>

Academic Editor: Mark J. Jackson

Received: 21 August 2023

Revised: 12 September 2023

Accepted: 16 September 2023

Published: 25 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of intelligent manufacturing and the acceleration of production rhythm, the requirements for “RAMS” (Reliability, Availability, Maintainability, and Safety) services are continuously upgrading. It is important to manage the health of industrial equipment and components efficiently [1,2]. However, traditional maintenance policies are incapable of responding promptly and accurately to the changes of system states that meet the demand for “RAMS” services. Therefore, with the support of real-time data acquisition and processing technologies, Predictive Health Management (PHM) solutions represented by Predictive Maintenance (PdM) has gradually become a current research hotspot under the industry 4.0 paradigm [3–5].

The PdM policy mainly involves two issues, which are state prediction and maintenance decision-making. Among state prediction methods, compared to expensive and

cumbersome physically based methods and empirically based methods with suboptimal accuracy [6,7], data-driven methods are a simple and convenient way that do not need to know the physical properties of the degradation mechanism and can get more accurate prediction results by selecting appropriate parameters [8]. Therefore, data-driven state prediction methods are more popular on the premise of enough reliability data [9–12]. The other maintenance decision-making issue considered in PdM policies is usually determining a maintenance plan that can optimize system performance according to the system state and certain indicators (cost, availability, reliability, etc.) [13–18], the core content of which is the formulation of decision-making rules and the optimization of the maintenance index or threshold value. As for the application of PdM, it is necessary to use the result of state prediction as the input for maintenance decision-making to form a complete and genuinely adequate PdM policy.

Data-driven prediction models have many advantages, but they lack of probabilistic explanation of the results, making it difficult to quantify the uncertainty of the prediction [19]. However, the uncertainty of changes in the state of equipment or components makes the state prediction methods used to characterize the uncertainty of expected results more reliable. It is also tough to calculate probability indicators, such as availability and reliability, by only using point estimates of degradation states to make maintenance decisions. Currently, only a small amount of research has been conducted to predict Remaining Useful Life (RUL) and quantify its uncertainty, which generally requires priori hypotheses [20,21], and the accuracy of the prediction results depends on the level of expert knowledge. In addition, there has been still few data-driven PdM approach that considers both state prediction and maintenance decision-making. It is incongruous to utilize RUL point estimation directly in maintenance decision-making that considering cost, availability, reliability, etc., which may be one of the main restrictive factors. The literature conducting maintenance decision-making research usually assumes that the degradation model is known [1,18] while there is often a lack of sufficient expert knowledge to identify the degradation model in advance actually, which leads to a significant limitation of the maintenance decision-making model. The paper by Nguyen et al. [1] seems to be the only study that considers RUL and its uncertainty prediction and maintenance decision-making simultaneously, but the model used is relatively simple, and only the probability of RUL in three different intervals can be obtained. The accuracy of the model needs to be improved. Given the above situation, the two main motivations for this work are as follows:

- **To fill the current research gap in RUL prediction and uncertainty quantification in data-driven PdM**, a new model to predict the dynamic continuous probability distribution of RUL without priori hypotheses needs to be established, which can improve the rationality and adaptability of RUL prediction results and support the implementation of maintenance decision-making;
- **To provide reference for establishing and applying a complete PdM policy that integrates state prediction and maintenance decision-making**, a multifactorial maintenance decision-making method needs to be constructed based on RUL prediction and uncertainty quantification. A complete PdM policy with favorable performance should be obtained and experimental verification should be conducted.

To address these issues, this paper combined the Long Short-Term Memory (LSTM) network with the Kernel Density Estimation with a Single Globally-optimized Bandwidth (KDE-SGB) method and established a new dynamic PdM (D-PdM) policy as a reference solution to obtain the continuous probability distribution of RUL based on the data-driven approach. A dataset of bearing verifies the feasibility of the proposed D-PdM policy. The major contributions of this paper are as follows:

- **For the prediction of RUL and the quantification of its uncertainty, a new RUL prediction model is established**, which uses a deep LSTM network to classify RUL. Further, the KDE-SGB method is adapted to convert the classification result into a continuous probability distribution. The distribution of RUL without priori hypotheses is obtained and supports for subsequent maintenance decisions.

- **The maintenance decision-making method is furtherly established based on the continuous probability distribution of RUL.** By introducing a comprehensive optimization target that considers the maintenance cost rate, system availability, and reliability simultaneously, the optimization of maintenance time is realized, and the recommended maintenance time is given. A complete PdM policy integrating state prediction and maintenance decision-making is ultimately formed.
- **The proposed complete PdM policy is validated through a bearing dataset and compared with several other policies.** The effect of different maintenance operation costs and durations on the model outcomes is explored. The proposed policy has been proven to have good predictive performance, which can significantly reduce maintenance costs and heighten the availability and reliability of equipment or components.
- **The proposed policy enriches the means of RUL prediction and its uncertainty quantification and provides a reference for the effective connection between RUL prediction and maintenance decision-making.**

The remainder of this paper is structured as follows. Section 2 introduces recent and related work on the data-driven PdM. Section 3 describes the problem studied briefly. Section 4 details the established D-PdM policy based on LSTM network and KDE-SGB method. Section 5 evaluates the D-PdM policy using a public bearing vibration dataset, compares it with several other policies, and explores the effect of maintenance operation cost and duration on model outcomes to prove its effectiveness. Section 6 summarizes the full text.

2. Literature Review

This section mainly focuses on the recent and related research on data-driven PdM policies.

Thanks to the rapid development of data acquisition technologies, in the field of data-driven PdM, deep learning algorithms, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), etc., are favored because of their significant advantages in the automated process of big data [1,22–25]. Among many deep learning methods, the LSTM neural network is one of the most widely used deep learning methods to track the system state because of the ability to learn and memorize long-term sequences [26,27]. Compared to the traditional RNN, it can effectively avoid the problems of gradient explosion and gradient disappearance [28,29].

There have been many studies on RUL prediction. Abdelghafar et al. [30] proposed a predictive approach based on the Enhanced Adaptive Guided Differential Evolution-optimized Support Vector Machine (EAGDE-SVM) to offer high RUL prediction accuracy. Different evaluation criteria of classification, prediction, and optimization aspects had been used to evaluate the EAGDE-SVM. Soualhi et al. [31] combined two RUL techniques, recursive and direct RUL estimation, to estimate the system RUL when dealing with the variability of degradation trends and unknown failure thresholds. Prognostic health indicators (HIs) were constructed and chosen to characterize the system's degradation trajectory. The ensemble of the derived RULs and their HI trajectories were fused to estimate the final RUL directly. Shutin et al. [32] proposed a hybrid approach to such prediction models involving the joint use of physics-based models of adjustable bearings and data-driven models for fast on-line prediction of their parameters. It had been tested on highly loaded locomotive traction motor axle bearings for consideration and prediction of their wear and RUL. Hesabi et al. [29] used the LSTM network to classify and predict the RUL of the components in the system to determine whether the components have failed. Although these methods have obtained good prediction results, they only provide point estimates of the state. However, it is difficult to use only point estimates to calculate probability indexes, such as availability and reliability, and it is more reasonable to treat state prediction as a probability problem due to the uncertainty of the future system state [33].

A good PdM policy should estimate not only the system state's mean value, but also its probability density function [34]. However, due to the lack of probabilistic orientation, data-driven methods usually cannot estimate the probability density function of states,

which makes it difficult to quantify the uncertainty of predictions [19]. Therefore, it is necessary to establish further an appropriate uncertainty quantification method to improve the data-driven method. Zhao et al. [20] obtained the estimated value of RUL through the CNN model. They gained different RUL Confidence Intervals (CIs) based on Gaussian distribution and quantile regression, respectively. Bracale et al. [35] developed two models based on time series and quantile regression to predict the probability distribution of RUL. Caceres et al. [21] established a probabilistic Bayesian recursive RNN to deal with epistemic uncertainty in forecasting and, at the same time, set the aleatoric uncertainty to satisfy the Gaussian distribution to obtain the point estimate of RUL and its CI of certain confidence level. Li et al. [36] proposed a novel Bayesian Deep Learning (BDL)-based framework to capture the combined effects of aleatoric uncertainty and epistemic uncertainty in RUL forecasting and adopted a sequential Bayesian boosting algorithm to unify the state transition and observation information through a single BDL model; a good RUL probability distribution prediction effect is finally achieved. Gao et al. [37] obtained a series of RUL prediction values through RNN, assuming that RUL satisfies the Gaussian distribution, and used a Multilayer Perceptron (MLP) to obtain the probability of different RUL. Li et al. [38] adopted a Just-in-time Learning (JITL) scheme to deal with the randomness of fault evolution and the diversity of degradation patterns. They developed a Randomized and Smoothed Gradient Boosting Decision Tree (RS-GBDT) model for the prediction of RUL and its CI. For multi-component systems, Tamssaouet et al. [39] proposed an online joint uncertainty quantification and model estimation method based on particle filtering and gradient descent for predicting the RUL of a system and its CI, which considers the interaction between components. As for multi-component systems, Nguyen et al. [40] used a combination of probabilistic models and deep regression neural networks to predict the component's RUL distribution and then used the system architecture information to deduce system reliability and the quantitative formula of system-level RUL uncertainty. In general, there are still few studies on RUL prediction and quantification of its uncertainty. Most of these existing studies use the Bayesian framework to update prior estimates or combine specific regression models to predict RUL distributions. Prior hypotheses are required in these studies, so the accuracy of the forecast results could be easily affected by the subjective factors of the initial assumptions. It is worth mentioning that some studies have been devoted to the determination of RUL's CIs [20,21], but for maintenance decision-making, a continuous probability distribution of RUL would be more useful. As in numerous studies on maintenance decision-making, the probability distribution of RUL is obtained through stochastic models [18,41]. The probability of failure before any time can be obtained and is not limited to the probability of failure in a specific interval. It is convenient to give a more accurate maintenance plan, including spare parts ordering, maintenance personnel arrangement, etc.

In addition, although both state prediction and maintenance decision-making are involved in PdM policies and only the combination of the two can form a complete PdM policy, the above studies have not continued to study maintenance decision-making based on RUL prediction to achieve the full functionality of PdM.

Through the above analysis, two deficiencies in the current PdM-related research can be clarified:

- **There are still few studies on data-driven RUL prediction and its uncertainty quantification methods.** In the few existing studies, the scope of the prediction models used is extremely limited, and most of them need to make subjective priori hypotheses, making the model's accuracy highly dependent on prior knowledge. Further extensive research is required, especially the establishing predictive models that reduce the impact of subjective factors, including priori hypotheses.
- **There is still a lack of research to establish a complete PdM policy by considering both state prediction and maintenance decision-making.** It is difficult to align many state prediction methods that provide only RUL point estimates directly with maintenance decision-making methods that assume the distribution model of state

degradation as a known condition. Therefore, combining RUL and its uncertainty prediction with maintenance decision-making is necessary to obtain a complete PdM policy with good generality.

The purpose of this work is to establish a data-driven PdM policy that covers RUL continuous probability distribution prediction and maintenance decision-making to fill these gaps. To predict RUL and quantify its uncertainty, a new RUL prediction model based on deep learning algorithms without priori hypotheses is established. It allows for the prediction of the dynamic continuous probability distribution of RUL to gain a more comprehensive understanding of the operational state changes of equipment or components. A maintenance decision-making method corresponding to RUL prediction is developed simultaneously to form a complete PdM policy that combines state prediction and maintenance decision-making, providing a reference for the development and application of PdM functions.

The comparison between this paper and the existing research on RUL prediction is shown in Table 1.

Table 1. The comparison between this paper and existing research related to the Remaining Useful Life (RUL) prediction.

Related Research	State Prediction					Maintenance Decision-Making
	Prediction Method	Prior Knowledge Is Required	Point Estimate	Quantitative Form of Uncertainty		
				Confidence Interval (CI)	Continuous Probability Distribution	
Zhao et al. [20]	Convolutional Neural Network (CNN) and quantile regression	✓	✓	✓		
Bracale et al. [35]	Time series and quantile regression	✓	✓	✓		
Caceres et al. [21]	Probabilistic Bayesian recursive Recurrent Neural Network (RNN)	✓	✓	✓	✓	
Li et al.[36]	Bayesian Deep Learning (BDL) and sequential Bayesian boosting algorithm	✓	✓	✓	✓	
Gao et al. [37]	RNN and Multilayer Perceptron (MLP)	✓	✓	✓		
Li et al. [38]	Randomized and Smoothed Gradient Boosting Decision Tree (RS-GBDT)		✓	✓		
Tamssaouet et al. [39]	Particle filtering and gradient descent		✓	✓	✓	
Nguyen et al. [40]	Probabilistic models and deep regression neural networks	✓	✓	✓	✓	
Thoppil et al. [42,43]	Bayesian optimized Long Short-Term Memory (LSTM) network and bidirectional-LSTM network		✓			

Table 1. Cont.

Related Research	State Prediction					Maintenance Decision-Making
	Prediction Method	Prior Knowledge Is Required	Point Estimate	Quantitative Form of Uncertainty		
				Confidence Interval (CI)	Continuous Probability Distribution	
Pater et al. [44]	LSTM autoencoder and similarity-based matching		✓			
Rathore et al. [45]	Attention-based stacked bidirectional-LSTM network		✓			
Nguyen et al. [1]	LSTM network		✓	✓		✓
This paper	LSTM network and Kernel Density Estimation with a Single Globally-optimized Bandwidth (KDE-SGB)		✓	✓	✓	✓

3. Problem Description

This section describes the problem studied and provides some basic assumptions and related notations utilized in this study.

The performance of equipment or components in the workshop will gradually deteriorate over time, and maintenance at an appropriate time will effectively save maintenance costs, ensure production efficiency, and extend service life. Frequently checking the equipment or components during operation is not practical, but continuous monitoring can be achieved by using various sensors to collect data. Therefore, using these data to predict the status of equipment or components, and determine the optimal maintenance time through specific evaluation indicators based on the state prediction results is necessary. This paper makes the following assumptions around state prediction and maintenance decision-making:

- Data on equipment or components can be continuously collected, and some characteristics of these data will change significantly and even regularly with the degradation of equipment or components, which can be used for state prediction;
- There is enough historical data available for the training of the prediction model;
- During maintenance, first check and confirm whether the equipment or components have been failed, and the failure can only be found during maintenance;
- Preventive measures shall be taken if there is no failure of equipment or components. Otherwise, corrective measures shall be taken;
- The cost and time required for preventive and corrective actions are known, regardless of the difference between different failure types.

The symbols, parameters, and variables are listed as shown in the abbreviations section.

4. Dynamic Predictive Maintenance Policy Based on LSTM Network

Taking advantage of LSTM network's outstanding characteristics of system state tracking, this paper establishes a dynamic PdM policy. The recommended maintenance time was given through the prediction of RUL probability distribution, combined with the comprehensive optimization target. This policy is shown in Figure 1. The constructed LSTM classifier realized the classification of the current RUL. Then, according to the RUL classification results, the probability distribution of the RUL was converted based on a kernel density estimation method to form a continuous RUL probability distribution. The

optimization objective considered several factors and constructed a comprehensive index as the optimization objective of the algorithm model. The specific content of each step is described below.

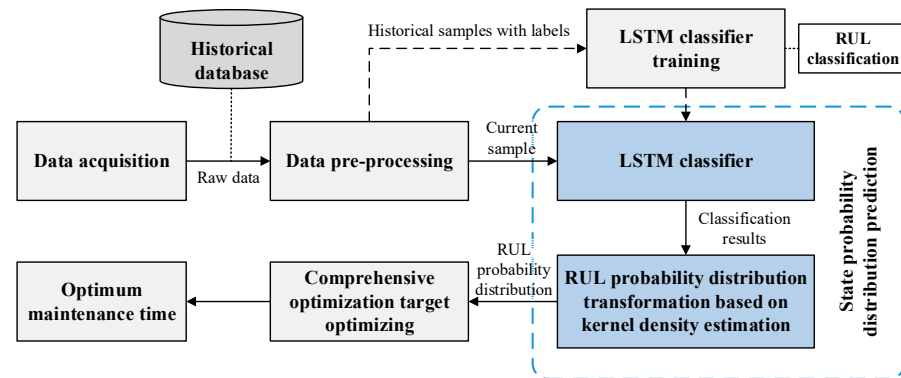


Figure 1. Dynamic Predictive Maintenance (PdM) policy based on LSTM network.

4.1. Data Pre-Processing

The raw data collected by sensors cannot be directly used for the input of the PdM model and must be formalized and labeled first.

Sample extraction. According to the input data format requirements of the model, the raw data was organized into samples in a unified form. Each sample in this paper was a two-dimensional tensor, including two dimensions of the time step and the sampling length of each time step. That is, from a series of consecutive data acquisition cycles of a specific number, each intercepted data segments of a specific length to form a sample, which corresponded to the state (RUL) of the last data acquisition cycle, as shown in Figure 2.

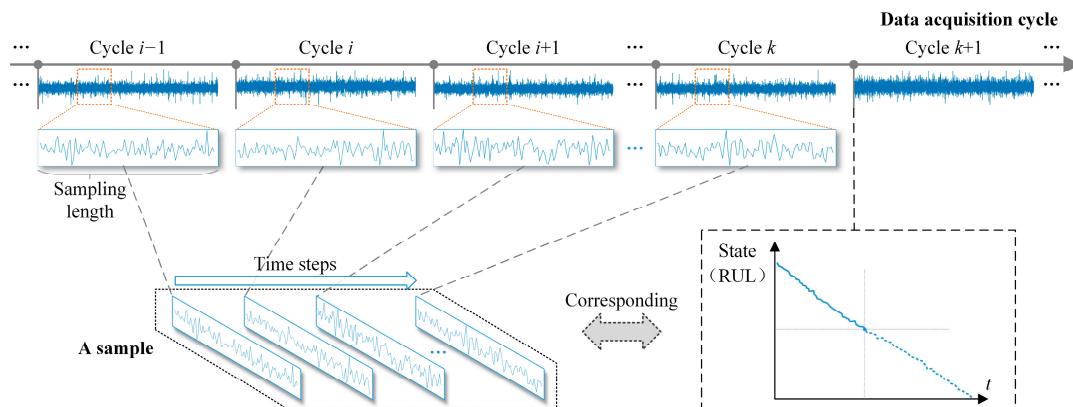


Figure 2. Sample extraction method. Data segments with a certain length over several consecutive data acquisition cycles are extracted to form a sample.

Sample labeling. To train a model and perform classification, the label was defined and assigned to each corresponding sample used for training. Labels could be defined by setting different RUL time windows. On the one hand, the classifier should not have too many categories. Otherwise, the classification effect will be poor. On the other hand, there should not be too few categories to facilitate the subsequent transformation of the RUL probability distribution. Because the system in the early stage was running normally and smoothly and there was still a long time before the failure occurs, there was no need to continue to pay attention to the RUL situation so that the early period could be regarded as the first category. The later period before failure could be divided into several categories.

Sample partition. Before training the LSTM classifier, the labeled samples needed to be divided into the independent training set, validation set, and test set. The training set was used for iterative training and continuous adjusting of the model parameters, and

the validation set was used to verify the generalization ability of the model during the iterative training process and to decide whether to stop training. When it needed to test the performance of the ultimate model obtained by training, the test set was used for testing.

4.2. LSTM Classifier

The structure of the constructed LSTM classifier is shown in Figure 3, which is divided into the input layer, the hidden layer, and the output layer. There were two LSTM layers in the hidden layer. Deep learning algorithms usually perform batch normalization during the data pre-processing step to improve the computing efficiency and the generalization ability of the overall model while the net input distribution of LSTM network neurons changes dynamically with time so the batch normalization method is not suitable. Therefore, layer normalization was utilized before each LSTM layer to effectively mitigate the situation of exploding or vanishing gradients and enhance the generalization ability of the model. After the two-layer LSTM network calculation, only the state output of the last time step was taken. Then, after two fully connected layers, the final classification calculation result was obtained, and the category corresponding to the largest item was used as the classification conclusion. The first fully connected layer used Relu as the activation function to keep the data dimension unchanged while the second fully connected layer used Softmax as the activation function to map the data dimension to the number of categories. To prevent over-fitting and improve the classifier effect, a dropout layer was set behind each LSTM layer and the previous fully connected layer.

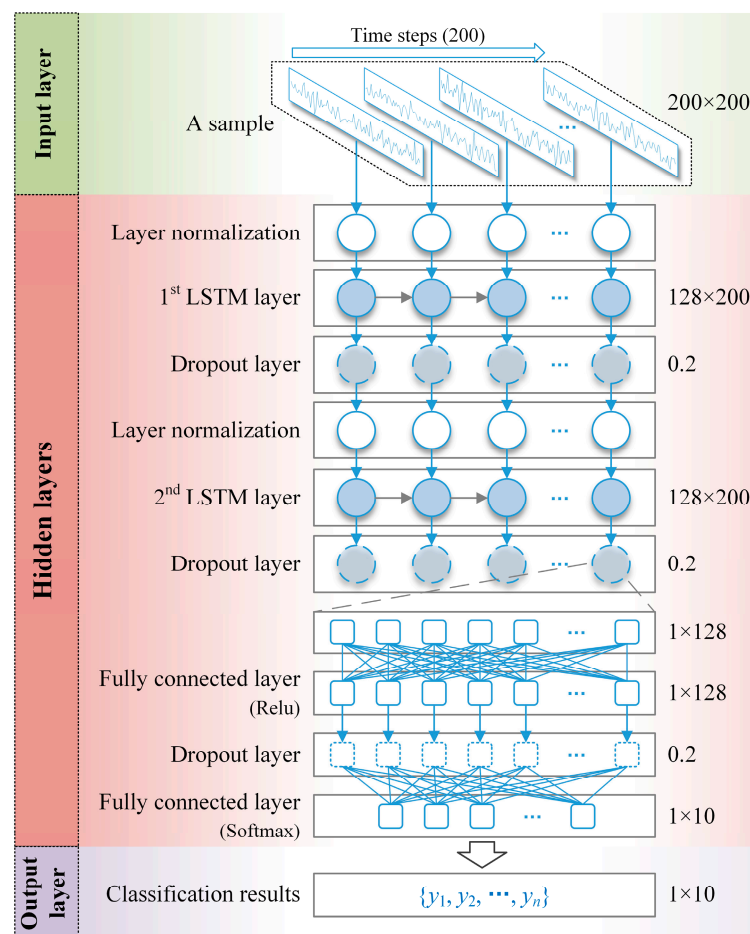


Figure 3. The structure of the LSTM classifier. The data on the right represents the sample dimensions in the input layer, the dropout rate of the dropout layer, and the feature dimensions in other network layers, respectively.

In practical applications, each parameter involved in the LSTM classifier could be adjusted according to the situation. It is worth noting that although the classification results of the output layer can be regarded as the possibility that RUL belongs to each category; such a discrete probability distribution is too “rough” to describe the specific situation of the RUL distribution. It is not enough to support the subsequent optimization of comprehensive targets and the recommendation of maintenance time. Therefore, further converting the RUL classification results into continuous probability distribution was imperative.

4.3. RUL Probability Distribution Transformation Based on KDE-SGB

Using the LSTM classifier can only obtain the probability of the corresponding state within RUL windows, and the distribution is discrete and with few elements. Therefore, converting the discrete probability distribution into a continuous and smooth probability distribution was considered by generating a series of random data points in a certain method within each category and conducting nonparametric estimation. The process of converting RUL classification results into a continuous probability distribution is shown in Figure 4, which is divided into five steps.

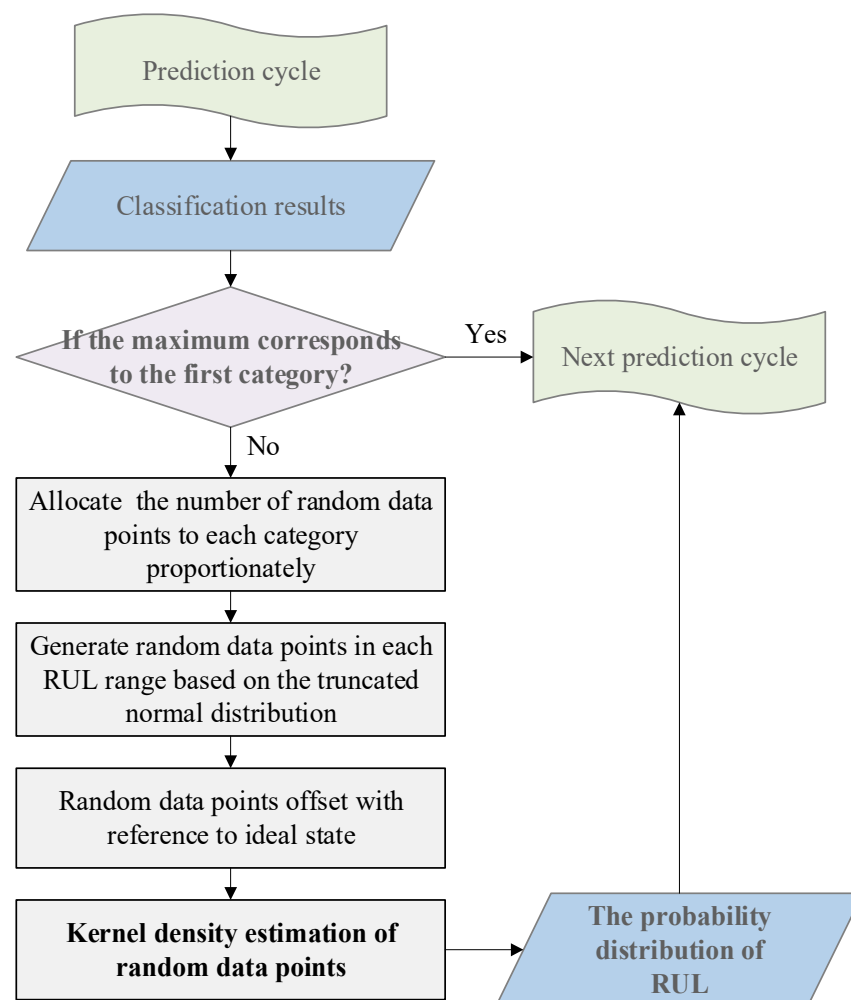


Figure 4. The process of RUL probability distribution transformation.

Step 1: Classification conclusion judgment

The maximum value of the classification result determined the category of RUL. In the early stage of normal and stable operation, that is, in the first category of RUL, there was no need to pay further attention to the probability distribution of RUL. Therefore, if the RUL belonged to the first category, it went directly to the next prediction cycle; otherwise, it proceeded to the following steps.

Step 2: Random data point number assignment

The given total number N of random data points was allocated to each category according to the proportion of the classification results. First, the total number N was multiplied, respectively, by each item in the classification result and rounded down to complete the preliminary allocation. Then, the fractional part of these multiplication results from largest to smallest was sorted, and one more data point was assigned to corresponding categories in the sequence until the remaining data points were all allocated.

Step 3: Random data points generation

A series of random data was generated in each RUL window according to the number of allocated random data points. Among them, the RUL window of the first category that was not cared about was replaced with a new one when calculating the distribution. The new window's width was the same or close to the other categories and was still adjacent to the RUL window of the second category. This adjustment was adopted to improve the accuracy of the probability distribution because under the premise that the classifier had a certain accuracy, these RULs belonging to the first category that were misidentified as other categories were also mainly concentrated in a small range close to the second category. Random data were generated within the RUL window for each category using a truncated normal distribution. The expectation value μ of the truncated normal distribution was obtained by averaging the sum of the products of the midpoint of each category's RUL window and the corresponding value in the classification results, and the standard deviation σ could be determined regarding the accuracy of the LSTM classifier. In this paper, $1/4$ width of the RUL window of the currently predicted category was taken as the σ to ensure that the randomly generated data points were always mainly distributed within the width of the RUL window, without excessive dispersion or concentration, thus avoiding the final probability distribution being not smooth enough.

Step 4: Random data points offset

The LSTM classifier does not have many categories and should not have too many, so it is easy to belong to the same category for a long time, and the classification results will not change much. At the same time, in the critical phase, when two categories switch, the classification results will transform rapidly. To avoid these two situations causing the probability distribution prediction result of RUL to stagnate for a long time or fluctuate too fast, the offset of random data points needed to be set appropriately. For all categories, the ideal expectations $\{I_1, I_2, \dots, I_n\}$ were set with the upper limits of RUL windows as the initial value. In the whole maintenance cycle, each time a prediction was executed to obtain the RUL classification result, one prediction cycle duration t_s was subtracted from the I_i of the related category. The difference between the updated I_i and expectation value μ was the offset value of the random data point.

Step 5: Kernel density estimation

Random data points were regressed using the KDE-SGB method proposed by Shimazaki et al. [46]. The basic principle was to form an overall probability distribution through the superposition of the kernel densities of a series of data points within a small interval, and the width of the globally unified interval (i.e., bandwidth) was optimized. The kernel density estimation method does not use prior knowledge about the data distribution and does not add any assumptions to the data distribution. It only needs to set the kernel function. It is a method to study the characteristics of the data distribution from the data sample itself. Therefore, compared with the prediction method of pre-assuming the distribution type based on experience, the kernel density estimation can be closer to the actual situation of the data sample and is universal. When the number of discrete points was abundant in kernel density estimation, the selection of different commonly used kernel functions had little influence on the results. Therefore, this paper directly selected the normal distribution as the kernel function of the KDE-SGB. The single global optimization bandwidth of the kernel function was determined based on the principle of minimizing the Mean Integrated Square Error (MISE) between the estimated rate and the unknown underlying rate [46]. Using the series of randomly generated data points as input, the

estimated probability density for any location within a given range can be obtained through kernel density estimation.

After the above steps, the RUL probability distribution could be transformed and outputted. The pseudo-code of the above procedure is shown in Algorithm 1.

Algorithm 1: RUL probability distribution transformation.

```

1: input:  $\{y_1, y_2, \dots, y_n\}; N; \{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\}; \{I_1, I_2, \dots, I_n\}; t_s; \{t_1, t_2, \dots, t_m\};$ 
2: output:  $\{f(t_1), f(t_2), \dots, f(t_m)\};$ 
3: //Step 1:
4: if  $y_1 = \max\{y_1, y_2, \dots, y_n\}$  then
5:   go to the next prediction cycle;
6: else
7:   //Step 2:
8:   for  $i \leftarrow 1$  to  $n$  do
9:      $S_i \leftarrow \lfloor N \cdot y_i \rfloor;$ 
10:     $D_i \leftarrow N \cdot y_i - S_i;$ 
11:    sort  $\{D_1, D_2, \dots, D_n\}$  from the largest to the smallest, and mark the subscript sequence after sorting as  $\{k_1, k_2, \dots, k_n\};$ 
12:    for  $j \leftarrow 1$  to  $(N - \sum_{i=1}^n S_i)$  do
13:       $S_{k_j} \leftarrow S_{k_j} + 1;$ 
14:   //Step 3:
15:    $\mu \leftarrow \sum_{i=1}^n (\frac{a_i + b_i}{2} \cdot y_i) / n;$ 
16:   for  $i \leftarrow 1$  to  $n$  do
17:      $\sigma \leftarrow (b_i - a_i) / 4;$ 
18:     randomly generate  $S_i$  number of data points through truncated normal distribution  $g(x|\mu, \sigma, a_i, b_i)$ , denote as  $P_i;$ 
19:    $P \leftarrow \bigcup_{i=1}^n P_i;$ 
20:   //Step 4:
21:    $(I_i \leftarrow I_i - t_s) | (y_i = \max\{y_1, y_2, \dots, y_n\});$ 
22:    $d \leftarrow (I_i | (y_i = \max\{y_1, y_2, \dots, y_n\})) - \mu;$ 
23:    $Q \leftarrow \{p + d | p \in P\};$ 
24:   //Step 5:
25:   use the KDE-SGB method to obtain  $\{f(t_1), f(t_2), \dots, f(t_m)\};$ 
26:   return  $\{f(t_1), f(t_2), \dots, f(t_m)\};$ 

```

4.4. Comprehensive Optimization Target

Currently, the most common optimization target in PdM-related research is maintenance cost, including inventory cost of works-in-progress and spare parts, equipment downtime cost, etc. [14], but some methods of reducing maintenance costs often do not consider other parameters, such as equipment availability, total production cost, etc. In practice, these methods can negatively affect other goals while reducing maintenance costs [47]. Although cost is essential to production management, other factors cannot be ignored. Therefore, many scholars have also researched maintenance decision-making, considering equipment availability [48,49], reliability [50], and so on. To improve the universality of the proposed model, this paper comprehensively considered the three optimization targets of the maintenance cost, availability, and reliability of systems. It adopted a weighted Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) approach [51] to construct a comprehensive optimization target. The recommended maintenance time was given through optimization of it.

- **Maintenance cost rate calculation**

The costs of taking preventive and corrective actions (including inspection costs) were C_p and C_c , respectively. Downtime will generate additional costs; let the downtime cost per unit time be C_d . Then, the expected maintenance cost rate EC was:

$$EC = \frac{1}{Et_{\text{total}} - Et_d} (C_p(1 - P_f(t)) + C_c P_f(t) + C_d \cdot Et_d), \quad (1)$$

where t was the maintenance time, which meant maintenance would be performed after time length t . $P_f(t)$ was the probability that the RUL in current was less than t , i.e., the cumulative distribution value of the predicted RUL probability distribution. Et_{total} was the estimated total duration of the current maintenance cycle, and Et_d was the estimated downtime duration. When a failure occurs, the equipment needs to be shut down immediately for corrective replacement. Preventive maintenance of the device or component can be performed during idle time to avoid taking up equipment operating time and increasing downtime costs. However, the idle time and the degradation process of the device or component are independent of each other, and the degradation process is uncertain. Therefore, referring to the daily maintenance cycle, the system work cycle, and the time required to arrange the preparation work for maintenance (such as ordering spare parts and arranging maintenance personnel), etc., the time interval Δt was given. When RUL is less than Δt , it must also be shut down for maintenance to avoid failure. Assuming that the equipment or component has been running for a duration t_0 in the current maintenance cycle, and the time required for preventive maintenance and corrective replacement was t_p and t_c , respectively, then:

$$Et_{total} = t_0 + t + t_c P_f(t) + t_p (P_f(t + \Delta t) - P_f(t)) \tag{2}$$

$$Et_d = t_c P_f(t) + t_p [P_f(t + \Delta t) - P_f(t)] + \int_{-\infty}^t (t - x) f(x) dx, \tag{3}$$

where $f(x)$ was the probability density value of RUL. The first term in the Formula (3) was the downtime caused by corrective replacement when a failure occurs, the second term was the downtime caused by shutting down for maintenance when the RUL was less than Δt , and the third term was the downtime caused by the failure before the maintenance.

• **Availability calculation**

The availability represents the percentage of time that equipment or critical components can be normally running. The formula of its expectation EA was:

$$EA = \frac{Et_{total} - Et_d}{Et_{total}}. \tag{4}$$

• **Reliability calculation**

The reliability represents the probability that equipment or key components will not fail until a certain point in time. Its expectation ER could be expressed as:

$$ER = 1 - P_f(t). \tag{5}$$

• **Comprehensive optimization target**

For maintenance decision-making, the lower the maintenance cost, the better, and the higher the availability and reliability, the better. To facilitate the establishment of the comprehensive optimization target, the expectations of maintenance cost rate, availability, and reliability were homogenized and normalized, which were denoted by Z_C , Z_A , and Z_R , respectively. Generally, there were upper and lower limits for maintenance cost rate and lower limits for availability and reliability. Therefore:

$$\begin{cases} Z_C = \frac{EC_{min}}{EC}, & Z_C \in \left(\frac{EC_{min}}{EC_{max}}, 1 \right) \\ Z_A = EA, & Z_A \in (EA_{min}, 1) \\ Z_R = ER, & Z_R \in (ER_{min}, 1) \end{cases} \tag{6}$$

Among them, the lower limit EC_{min} of the maintenance cost rate was obtained from the minimum value of EC in Formula (1). The upper limit EC_{max} of the maintenance cost rate, the lower limit EA_{min} of the availability, and the lower limit ER_{min} of the reliability were

preset or obtained by calculating the corresponding extreme values from Equations (1), (4), and (5), respectively. According to the TOPSIS method, the optimal solution and the worst solution were:

$$Z^+ = (1, 1, 1) \quad (7)$$

$$Z^- = \left(\frac{EC_{\min}}{EC_{\max}}, EA_{\min}, ER_{\min} \right). \quad (8)$$

The Euclidean distances between the three normalization indicators and the optimal and worst solutions were calculated, respectively:

$$D^+ = \sqrt{(Z_C - 1)^2 \omega_C^2 + (Z_A - 1)^2 \omega_A^2 + (Z_R - 1)^2 \omega_R^2} \quad (9)$$

$$D^- = \sqrt{\left(Z_C - \frac{EC_{\min}}{EC_{\max}} \right)^2 \omega_C^2 + (Z_A - EA_{\min})^2 \omega_A^2 + (Z_R - ER_{\min})^2 \omega_R^2}. \quad (10)$$

Among them, ω_C , ω_A , and ω_R were the weights of the normalized maintenance cost rate, availability, and reliability, respectively, which could be set and adjusted manually to realize the intervention in the PdM decision-making process. The closeness of the three normalization indicators to the optimal solution was:

$$C_D = \frac{D^-}{D^- + D^+}. \quad (11)$$

C_D was the comprehensive optimization target, and the time t corresponding to its maximum value was the recommended maintenance time.

5. Case Verification and Performance Evaluation Based on Bearing Vibration Data

Taking the bearing vibration data set [52] provided by the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati as an example, the above method was experimentally verified. The acquisition scheme of bearing vibration data was to install four bearings on one shaft, keep the rotational speed and radial load constant, and install the accelerometer on the bearing seat to collect vibration data. The verification content consisted of three parts: the accuracy of the LSTM classifier, the error of the RUL probability distribution, and the performance of the D-PdM. In addition, the influence of different maintenance operation costs and durations on the model results was further explored. The algorithm model was developed with python3, and the LSTM classifier was built under the TensorFlow machine learning framework. Datasets, programs, and calculation results of this work can be found in <https://www.kaggle.com/datasets/shulian00/pdm-bearing> (accessed on 17 September 2023).

5.1. Accuracy Verification of the LSTM Classifier

The data of the second experiment bearing 1 in the data set was selected for the training and verification of the classifier. The variation of the standard deviation of bearing vibration data is shown in Figure 5. It was assumed that the bearing was determined to be failed when the standard deviation reached a certain value and rose continuously. The entire process could be divided into three stages, the steady running stage, the significant degradation stage, and the failure stage. In the steady running stage, the bearing was in good condition, and there was no need to pay attention to its RUL distribution, so this stage was classified as category 1. In the significant degradation stage, the RUL of the bearing needed to be continuously concerned. For the subsequent RUL probability distribution transformation steps, the more RUL categories, the more accurate the distribution model will be. However, the number of categories should not be too large; otherwise, the accuracy of the classifier will be greatly affected. This paper divided the significant degradation

stage into eight parts, which were classified as categories 2~9. The last failure stage was classified as category 10.

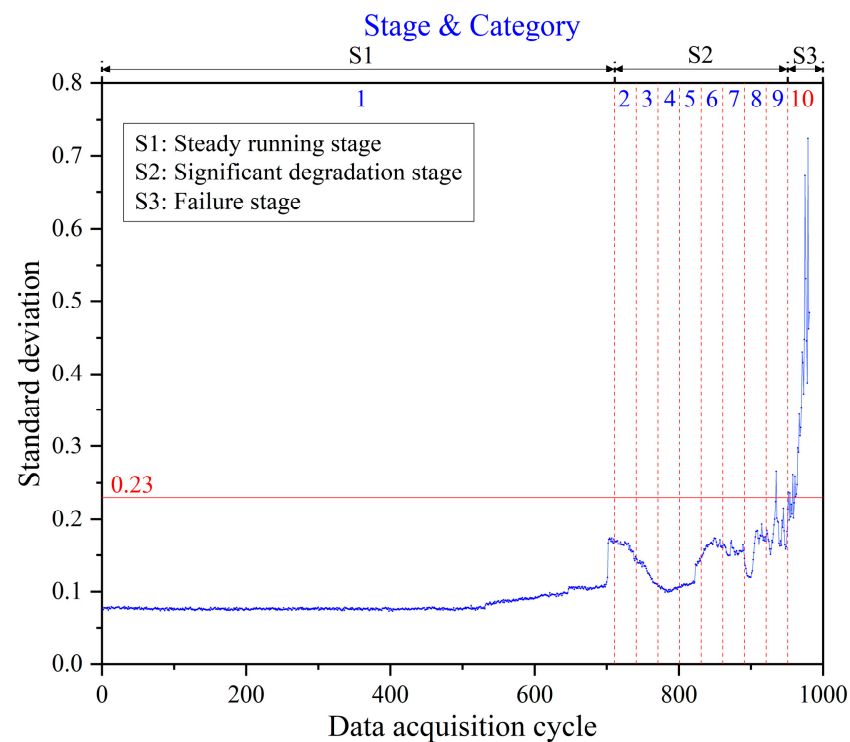


Figure 5. Range setting of RUL categories based on the standard deviation of vibration data.

The parameter settings of the LSTM classifier in this paper are shown in Figure 3. The dimensions of each sample were [200, 200], the hidden dimensions of each LSTM layer were 128, and the dropout rate was set to 0.2. The number of samples in the training, validation, and test sets was 16,000, 2000, and 2000, respectively. To keep the number of samples of each category balanced during training and verification, so as not to affect the validity of the model and the credibility of the verification results, the sample numbers of the three sample sets were evenly distributed to each category before the training samples were extracted, and in each category, they were further evenly distributed to each data acquisition cycle. If there were still remaining parts, they were distributed randomly. The data of each data acquisition cycle was divided according to the ratio of 8:1:1 and used to extract training samples, validation samples, and test samples, respectively. When extracting samples, according to the aforementioned sample extraction method, 200 vibration data were extracted in 200 consecutive time steps (i.e., data acquisition cycles), thereby forming samples with dimensions [200, 200]. The category of the last data acquisition cycle in each sample was the category of the sample, and the sample label was assigned according to the category. The samples extracted by the above method formed the training set, the verification set, and the test set.

Then, the training set was imported into the built LSTM classifier model for training. The samples were shuffled before training to avoid too many samples of the same category in a batch, which would significantly impact the model's accuracy. The batch size for training was set to 100, the optimizer used Adam, the learning rate was set to 1×10^{-3} , and the loss function was specified as a categorical cross-entropy loss function. The total number of epochs was set to 300. After each iteration epoch, the classification accuracy was monitored through the validation set. The training would be terminated early if the model's accuracy did not improve for 50 consecutive epochs. After three cycles of training, the best accuracy rates monitored by the validation set were 96.85%, 97.00%, and 96.75%, respectively. The best model obtained from the second training with the highest accuracy rate was selected and verified by inputting the validation set.

The accuracy of the classifier was evaluated using the probability confusion matrix [1]. The probability confusion matrix of the classification conclusion after importing the test set is shown in Figure 6. The overall accuracy rate of the test set classification was 96.95%. Except for the fourth category, where the accuracy rate was below 95% and it was relatively easy to predict RUL as the latter category incorrectly, the prediction accuracy of other categories was above 95%, especially in the last two categories around the bearing failure, the accuracy remained high, which was of great help to the PdM of the bearing. The case of classification prediction errors is mainly about predicting the RUL as a category adjacent to the actual category, and these errors should mainly occur at the junction between the ranges of different categories, which is a normal phenomenon.

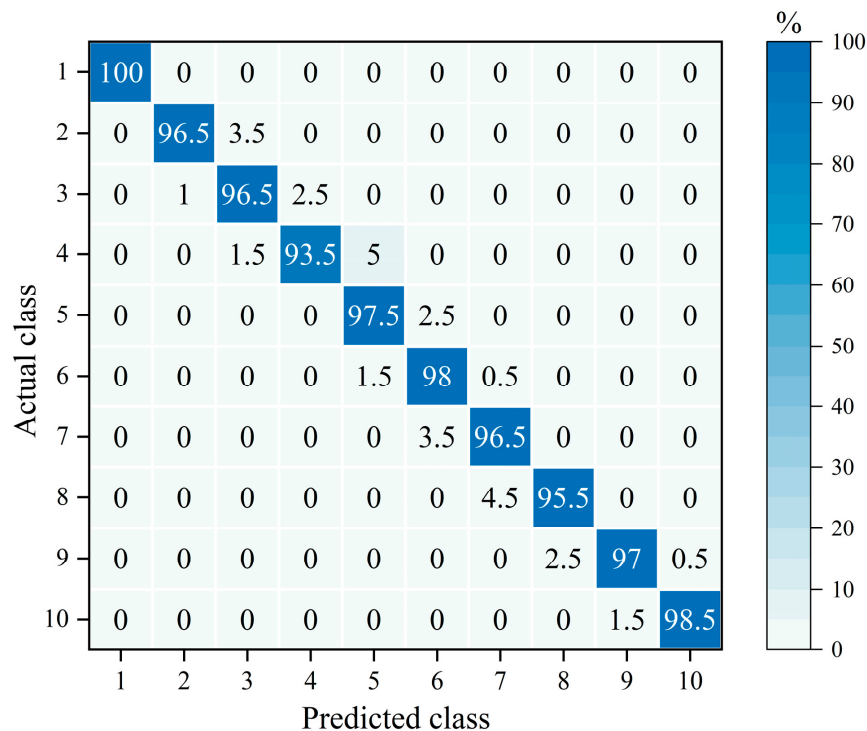


Figure 6. The probability confusion matrix of the second experiment bearing 1’s test set classification conclusion.

To test the performance of the LSTM classifier on different bearings, another bearing (the third experiment bearing 3 in the data set) was selected for model training and verification, according to the previous steps. The probability confusion matrix obtained from classification verification is shown in Figure 7. The overall accuracy rate of the test set classification was 86.45%, and the performance was not very good in the 3rd, 4th, and 5th categories, in which the accuracy rates were below 80%. It is relatively easy to predict RUL as the first few categories incorrectly, which is due to the bearing in the early stage maintaining a stable operation state for a long time, and its vibration signal characteristics also remain almost unchanged for a long period. In contrast, the prediction accuracy of the categories from category 7 onwards was above 95%, maintaining a high level, which showed that the model can still accurately grasp the state change trend of the bearing in the later stage of operation. Accurate prediction of the state close to failure is the key to bearing maintenance, and the low prediction accuracy during the early steady running stage has little impact on the maintenance activity plan. Of course, in practical applications, the number of parallel prediction channels in each prediction cycle can also be increased, and the final classification conclusion can be determined by comparing the prediction results of all channels to reduce the risk of the wrong prediction further.

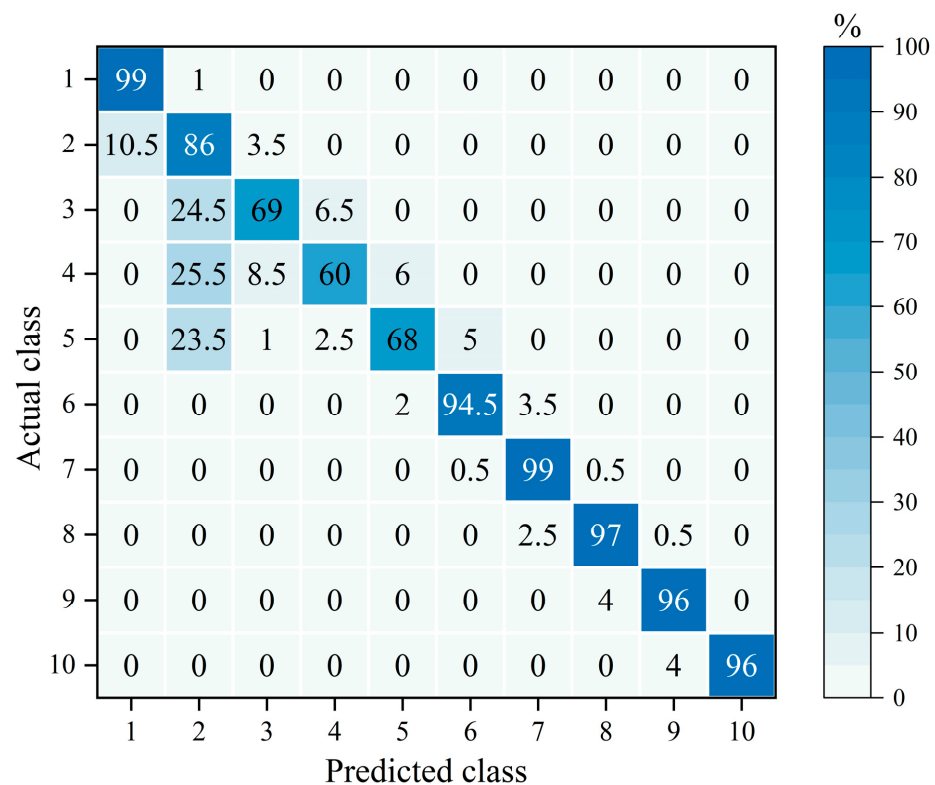


Figure 7. The probability confusion matrix of the third experiment bearing 3's test set classification conclusion.

The above test shows that the constructed LSTM classifier has high accuracy and certain portability.

5.2. Error Analysis of RUL Probability Distribution

After the classification of the RUL was completed, the continuous probability distribution of the RUL was obtained by further conversion based on the classification result. In this paper, the number N of random data points in the transformation process of RUL probability distribution in each prediction period was 10,000. The probability density values at 101 equidistant time points were gained for the subsequent calculation of the comprehensive optimization target. These time points were taken within the significant degradation stage and the 30 data acquisition cycles before and after the significant degradation stage (including two endpoints) for the convenience of obtaining the RUL probability distribution profile at both ends of the significant degradation stage. This was also the concerned stage of the whole degradation process.

To evaluate the error of the resulting RUL probability distribution, appropriate extraction of the vibration data was required to “obtain” data for complete degradation cycles. The test set and validation set of the second experiment bearing 1 in Section 4.1 was selected, and the data of each data acquisition cycle were divided into data segments with a length of 200. Then, a data segment in each data acquisition cycle was randomly selected in turn to form a degeneration cycle. By analogy, three degradation cycles were extracted without repetition and used for error analysis of the RUL probability distribution.

The data of the three degradation cycles were classified with the LSTM classifier and converted into the probability distribution with the kernel density estimation, respectively, and the RUL probability prediction of the concerned stage in each degradation cycle was obtained as shown in Figure 8. It can be seen from the figure that, at different times, the RUL probability distribution is always concentrated in a small area, and its probability density decreases rapidly on both sides. The whole RUL probability distribution shifts to the smaller side with the decrease in the actual RUL value. What is more, the continuous

probability distribution of RUL at any time can be obtained during maintenance decision-making. In the three degradation cycles, the RUL probability distribution has obvious periodic changes. Near the position of the transition between the two categories, the peak value of the RUL probability distribution is significantly lower than that of the prediction cycles before and after. This is caused by the rapid conversion of the probabilities around the boundary of two categories when making classification by the LSTM classifier, making it more difficult to predict the actual situation of RUL accurately in this region compared to other regions. It is worth mentioning that, since the RUL distribution of the system is predicted through the collected data, to facilitate the model calculation, there is a region with a negative RUL value, which indicates the probability of failure. The RUL probability distribution was further analyzed through the Root Mean Square Error (RMSE) and the mean deviation probability within a certain range, as shown in Table 2. The RMSE values of different degradation cycles have obvious fluctuations, but they are all kept within a small range. As for the deviation of the mean of the RUL probability distribution from the actual RUL, within the range of ± 5 , the first degradation cycle with the lowest probability can still be maintained above 80%. The deviations of all the mean values are within the range of ± 10 . The mean value of RUL probability distribution and its 90% and 95% CI are shown in Figure 9, indicating that the changing trend of RUL distribution probability concentration area is consistent with the change of actual RUL, and the accuracy of RUL prediction results maintains a high level. This result shows that, compared to the classifier only obtaining the discrete probabilities of several categories, further probability distribution transformation greatly improves the level of RUL prediction.

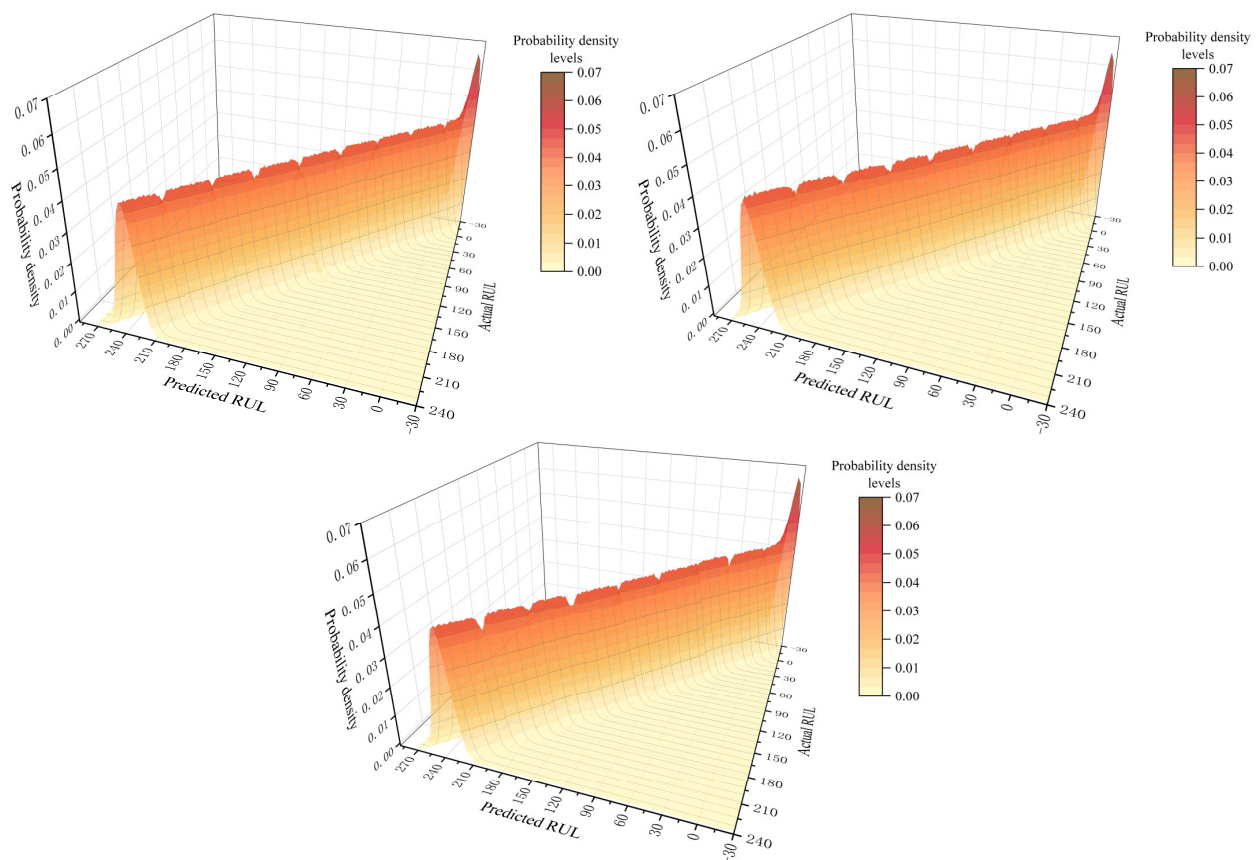


Figure 8. RUL probability distribution for the concerned stage in the three degradation cycles.

Table 2. RUL probability distribution analysis.

Degradation Cycle	Root Mean Square Error	The Probability of Deviation within ± 5	The Probability of Deviation within ± 10
1	3.6036	80.74%	100.00%
2	2.2166	98.89%	100.00%
3	3.1606	89.63%	100.00%

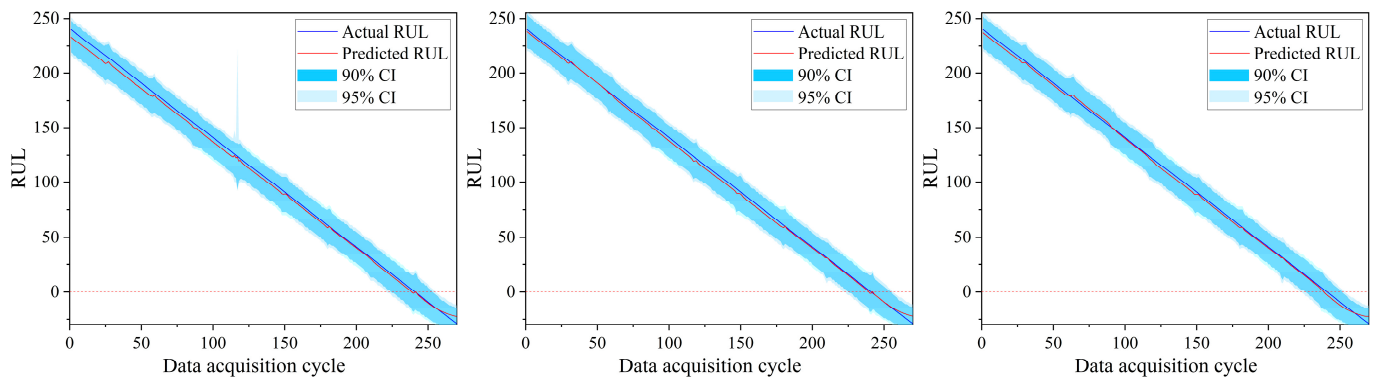


Figure 9. The mean value of the RUL probability distribution and its 90% and 95% CI for the concerned stages of the three degradation cycles.

5.3. Performance Evaluation of PdM Policies

To evaluate the performance of the D-PdM proposed in this paper, using the test set and validation set of the second experiment bearing 1 and the third experiment bearing 3, 40 degradation cycles were extracted and formed following the method in the previous section. The data collection interval was taken as the unit duration. Considering that the difference between the total duration of the two experiments was too great and the service life was also random in the actual situation, only the last 1000 data acquisition cycles of the third experiment bearing 3 were retained in order to be closer to the actual situation. A series of integers in the range $[-200, 200]$ were randomly generated as the runtime offset for the resulting series of degradation cycles. On this basis, the sequence of the extracted 40 degradation cycles was shuffled, so the degradation cycles' duration was variable and random. It was assumed that the corresponding model parameter set could be correctly selected when using the LSTM classifier for classification.

During performance evaluation, Periodic Maintenance (PeM) policy, Classification-based Predictive Maintenance (C-PdM) policy, and Ideal Maintenance (IdM) policy were used as comparisons.

- **Periodic Maintenance (PeM) policy**

The time point t_{PeM} for PeM was calculated from historical degradation cycle data, such that the ratio of the number of degradation cycles that fail before t_{PeM} to fail after t_{PeM} was closest to the ratio of the preventive action cost C_p to the corrective action cost C_c , where t_{PeM} took an integer. Since periodic maintenance can make full preparations for maintenance, such as ordering spare parts and arranging maintenance personnel, maintenance operations were considered being performed during idle time. In addition to the cost of preventive or corrective action and the cost of downtime due to corrective replacement, only the cost of additional downtime due to failure before maintenance was considered. The cost of a single maintenance cycle was:

$$C_{PeM} = \begin{cases} C_p, & t_{PeM} \leq L \\ C_c + (t_{PeM} - L + t_c)C_d, & t_{PeM} > L' \end{cases} \quad (12)$$

where L was the actual useful life corresponding to the current maintenance cycle. Since the failure of the system can only be known when maintenance is carried out, there are cases where failure has occurred, but maintenance is carried out after some time.

- **Classification-based Predictive Maintenance (C-PdM) policy**

This policy also uses the LSTM classifier established in this paper but does not perform subsequent probability distribution conversion, directly regards the classification results as the probability of each category instead, and the maintenance decision is determined by judging whether the probability of failure exceeds the threshold and by comparing the cost rates of immediate maintenance and no maintenance at present. Since it was impossible to make a clear judgment on the RUL, which made it difficult to prepare in advance, to avoid failures in the maintenance process, the equipment was immediately stopped to execute maintenance operations (it can be seen from the subsequent numerical experiment results that there was indeed a considerable probability of failure if maintenance was selected to be carried out at the idle time). The threshold of failure probability was set to 1%. Immediate maintenance in the current prediction cycle will incur preventative maintenance cost C_p and downtime duration t_p , and no maintenance will risk failure in the next prediction cycle. The cost rate expectations EC_{DR} and EC_{DN} for immediate maintenance and no maintenance temporarily were:

$$EC_{DR} = \frac{C_p + t_p C_d}{t_0} \quad (13)$$

$$EC_{DN} = \frac{y_{10}(C_c + t_c C_d) + (1 - y_{10})(C_p + t_p C_d)}{t_0 + 1}, \quad (14)$$

where y_{10} was the value of the last category in the classification result, regarded as a probability. By judging whether y_{10} exceeded 1% and comparing the magnitudes of EC_{DR} and EC_{DN} , the option with a lower expected cost rate was selected. In practice, additional downtime costs are incurred if a failure has already occurred before maintenance is carried out. The actual cost of a single maintenance cycle was:

$$C_{CPdM} = \begin{cases} C_p + t_p C_d, & t_{CPdM} \leq L \\ C_c + (t_{CPdM} - L + t_c) C_d, & t_{CPdM} > L' \end{cases} \quad (15)$$

where t_{CPdM} was the maintenance time point determined by the C-PdM policy.

- **Ideal Maintenance (IdM) policy**

IdM is a hypothetical perfect maintenance policy that enables optimal maintenance at the optimal time point, that is, at the last moment before failure, and completes preventive maintenance in idle time without incurring downtime costs. Each maintenance cycle coincides with the degradation cycle, and the cost of each maintenance cycle is equal to C_p . This is not achievable in practice and is only used as a reference for other policies in this paper.

- **Dynamic Predictive Maintenance (D-PdM) policy**

In the D-PdM policy in this paper, considering that the prediction results might still lead to misjudgments, for the sake of safety, when the recommended maintenance time given twice in a row was less than or equal to 5, the maintenance operation was arranged, denoted as $\delta = 0$. When the recommended maintenance time is 0, it will stop immediately for maintenance, which is recorded as $\delta = 1$. In practice, if a failure has occurred before the maintenance operation is scheduled, the corrective replacement cost and downtime costs are incurred. If the maintenance operation is scheduled within 5 cycles before the failure occurs without immediate shutdown, the bearing is considered having

failed during maintenance scheduling and requires corrective replacement. The cost of a single maintenance cycle was:

$$C_{DPdM} = \begin{cases} C_p, & \delta = 0 \& (t_{DPdM} + 5) \leq L \\ C_p + t_p C_d, & \delta = 1 \& t_{DPdM} \leq L \\ C_c + t_c C_d, & \delta = 0 \& t_{DPdM} \leq L < (t_{DPdM} + 5) \\ C_c + (t_{DPdM} - L + t_c) C_d, & t_{DPdM} > L \end{cases} \quad (16)$$

The variable t_{DPdM} was the maintenance time point determined by the model.

The above policies were compared through the data of 40 degradation cycles, and the actual overall maintenance cost rate, overall availability, and overall reliability under different policies were compared. Here, the overall reliability was calculated by the percentage of maintenance cycles without failures. The values of the relevant parameters in this paper are shown in Table 3, where t_{PeM} was calculated from the training set data. The obtained results are shown in Table 4, and the performance comparison of these policies is shown in Figure 10. It can be seen that except for 11 cycles of failures under the PeM policy, no failures occur under other policies. Due to failures and premature stops of maintenance cycles, PeM has longer downtime and shorter operating duration, resulting in significantly shorter total maintenance cycles duration and total normal operating duration compared to D-PdM and C-PdM. The total maintenance cycles duration and total normal operating duration of D-PdM and C-PdM are not significantly different from IdM. In terms of overall maintenance cost rate, PeM is the highest, reaching 0.9474. Compared with PeM, D-PdM and C-PdM save 70.02% and 60.30% in maintenance costs, respectively. In terms of overall availability, the D-PdM and C-PdM maintain high levels, approaching 100%, and significantly higher than that of PeM. The overall availability of D-PdM is 0.46% higher than that of C-PdM. In terms of overall reliability, due to some failures in the PeM policy, the reliable operation of the bearings was not well guaranteed, with a reliability of only 72.50%. While D-PdM and C-PdM can quickly issue warnings and ensure stable and reliable operation of bearings. It can be concluded that compared to PdM, D-PdM and C-PdM can comprehensively improve the health management level of critical equipment or components. Overall, the performance of D-PdM is superior to that of C-PdM, and it is very close to IdM under ideal conditions. This is because D-PdM can obtain the RUL distribution of bearings more precisely at any time, especially when approaching failures, to more accurately determine whether maintenance is needed, maximize the service life of bearings, and avoid failure risks. Compared to PeM and C-PdM, D-PdM can reduce maintenance costs by 24.49~70.02%, improve availability by 0.46%~1.90%, and improve reliability by 0.00~27.50%. Therefore, it is believed that the D-PdM policy proposed in this paper has good performance for the bearing case used.

Table 3. The values of the relevant parameters in the performance evaluation of PdM policies.

Parameter	C_p	C_c	C_d	Δt	t_p	t_c	ω_C	ω_A	ω_R	t_{PeM}
Value	250	1000	20	5	5	20	0.6	0.2	0.2	847

Table 4. Maintenance results under different policies.

Policy	Number of Failures	Total Maintenance Cycles Duration	Total Normal Operating Duration	Total Maintenance Cost	Overall Maintenance Cost Rate	Overall Availability	Overall Reliability
PeM	11	34,100	33,429	31,670	0.9474	98.03%	72.50%
C-PdM	0	37,425	37,225	14,000	0.3761	99.47%	100.00%
IdM	0	38,407	38,407	10,000	0.2604	100.00%	100.00%
D-PdM	0	36,998	36,973	10,500	0.2840	99.93%	100.00%

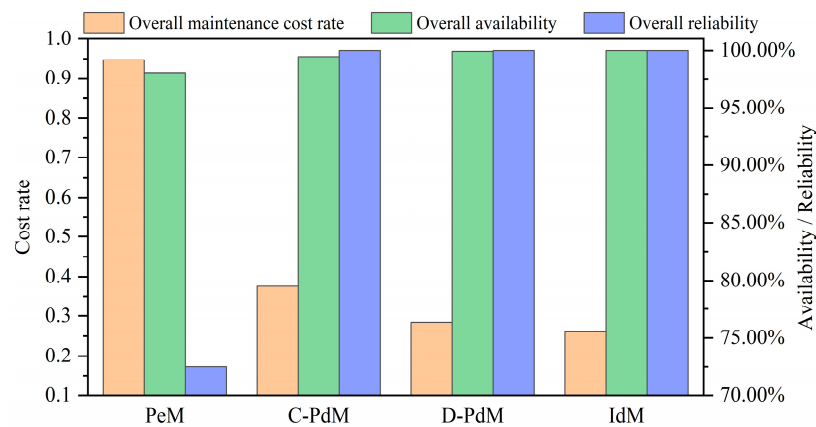


Figure 10. Performance comparison of different maintenance policies.

5.4. Influence of Different Maintenance Operation Costs and Durations

For maintenance policies, the different costs and durations of maintenance operations will have a significant influence on the maintenance schedule. Several situations that may arise in practice are considered. (1) The system cannot be subjected to preventive maintenance and can only be replaced, that is, $C_p = C_c$ and $t_p = t_c$. (2) C_p is much less than C_c . (3) C_p is less than C_c , but the difference in the order of magnitude between the two is small. (4) The variable t_p is much less than t_c . (5) The variable t_p is less than t_c , but the difference in the order of magnitude between the two is small. According to the above situation, the working conditions for setting different values of C_p , C_c , t_p , and t_c are shown in Table 5, and other parameters remain unchanged.

Table 5. Different maintenance operation costs and durations.

Parameter	C_p	C_c	t_p	t_c
Case 1	1000	1000	20	20
Case 2	1	1000	5	20
Case 3	250	1000	5	20
Case 4	1	1000	1	100
Case 5	250	1000	1	100

The overall maintenance cost rate, overall availability, and overall reliability of C-PdM and D-PdM under various working conditions are shown in Figure 11. In general, the maintenance effect of D-PdM is better than that of C-PdM under all the different working conditions. In terms of maintenance cost rate, maintenance execution cost is the most important factor for the model results of C-PdM and D-PdM, which is consistent with common sense. Compared with C-PdM, the cost advantage of D-PdM is more reflected when the difference between C_p and C_c is larger, and the change is smaller when the maintenance time is different. In terms of system availability, it mainly depends on the size of t_p since both policies are effective in avoiding failure. Compared with the C-PdM policy, the D-PdM policy can more accurately grasp the system RUL at any time, which is more helpful to reduce downtime, so the system availability is higher under the D-PdM policy. As for the system reliability, in case 1, two maintenance cycles failed under the C-PdM policy, which is caused by the error in the classification result of the LSTM classifier. Even if the accuracy of the LSTM classifier can reach more than 96% within the period around the failure and y_{10} is limited to no more than 1%, it still cannot completely avoid the failure, indicating that the C-PdM policy has certain defects and may not be suitable when C_p and C_c is close to or even equal to each other. In contrast, the D-PdM policy ensures the safety of the system under all working conditions.

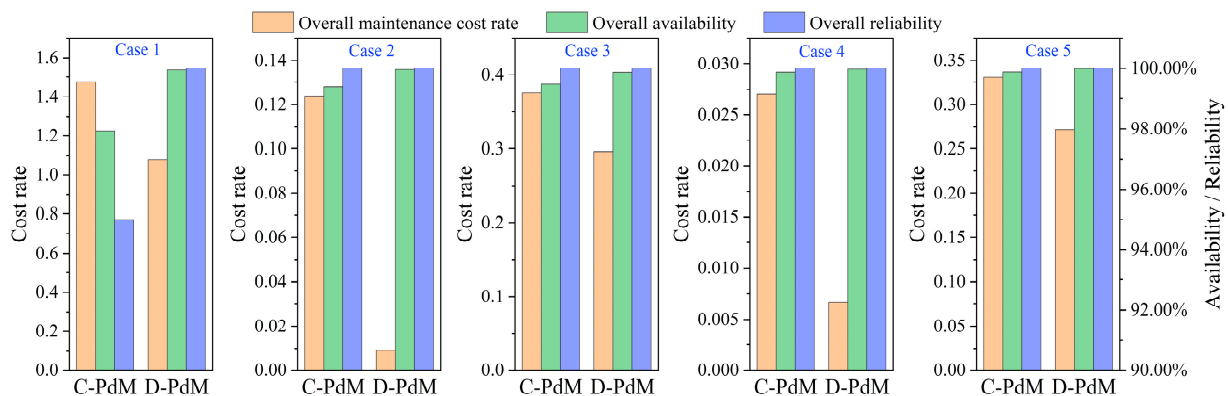


Figure 11. Comparison of maintenance effect between Classification-based Predictive Maintenance (C-PdM) policy and Dynamic Predictive Maintenance (D-PdM) policy under different working conditions.

Of course, the bearing data in the application case of the proposed policy are collected under stable operating parameters, but the actual conditions are constantly changing, so the performance of the policy under changing conditions remains to be verified. The proposed probabilistic transformation idea based on kernel density estimation is applicable to not only LSTM network-based state prediction methods, but also other data-driven methods. The PdM policy, including state prediction and maintenance decision-making in this paper, can provide a reference for constructing and applying the PdM function of equipment or components.

6. Conclusions

In data-driven PdM, there is still a lack of research on the uncertainty characterization of state prediction results and their application in integration with maintenance decision-making. This paper proposes a dynamic PdM policy to solve these problems, achieving the dynamic prediction of continuous probability distribution of RUL, and the maintenance decision-making with a comprehensive optimization target. Finally, a case study and performance evaluation of this policy are presented.

A new dynamic PdM policy is designed and developed using the LSTM network and kernel density estimation method. Through LSTM network-based RUL classification and RUL probability distribution conversion based on KDE-SGB, the continuous RUL probability distribution is obtained. The weighted TOPSIS method is used to construct a comprehensive optimization target according to the maintenance cost rate, availability, and reliability, and the dynamic maintenance decision is made based on this target. Finally, a public bearing vibration dataset is used to verify and evaluate the LSTM classifier, the RUL probability distribution conversion model, and the overall maintenance performance of the D-PdM policy. The proposed policy exhibits good performance in predicting the continuous probability distribution of RUL, with prediction results concentrated within a range of ± 10 data acquisition cycles. The D-PdM policy is compared with the PeM, C-PdM, and IdM policies, and the influence of different maintenance operation costs and durations on the maintenance effect is further explored. The outcomes show that the D-PdM policy constructed in this paper has comprehensive performance advantages in maintenance cost reduction and availability and reliability improvement. Compared with the two policies, PeM and C-PdM, the proposed policy can reduce maintenance costs by 24.49~70.02%, increase availability by 0.46~1.90%, and improve reliability by 0.00~27.50%. Moreover, the performance of it is more stable under various maintenance costs and durations.

The proposed policy provides a new approach for predicting RUL and quantifying its uncertainty. Compared with the method that only provides point estimation of the state, it is more conducive to comprehensively control the evolution of equipment or component operating states. It is also more beneficial for maintenance decision-making to consider probability indexes without the need for priori hypotheses and reducing the demand for

expert knowledge. The proposed policy links RUL prediction with maintenance decision-making, forming a complete PdM policy and providing reference cases for the construction and application of PdM. In addition, the RUL probability distribution transformation method proposed in this paper based on the KDE-SGB can also provide a continuous probability distribution transformation idea for the existing methods merely obtaining RUL point estimates so as to carry out maintenance decision-making better.

At present, the proposed policy is only tested with the bearing case under fixed operating conditions, demonstrating its effectiveness under a single operating condition, while its performance applied in complex operating conditions and for different equipment or components still needs to be verified. In the future, the application scenarios of the proposed policy will be expanded upon, and the systematic integration of PdM policies in the case of multiple components will be studied based on this work.

Author Contributions: Conceptualization, S.X. and W.Z.; Methodology, S.X.; Software, S.X.; Validation, S.X. and F.X.; Formal Analysis, S.X.; Investigation, S.X.; Resources, S.X. and W.Z.; Data Curation, S.X. and F.X.; Writing—Original Draft Preparation, S.X.; Writing—Review & Editing, F.X. and J.Z.; Visualization, S.X.; Supervision, F.X. and J.Z.; Project Administration, W.Z.; Funding Acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China [grant number 2022YFE0114100] and the National Key R&D Program of China [grant number 2017YFE0101400].

Data Availability Statement: Datasets, programs, and calculation results of this work can be found in <https://www.kaggle.com/datasets/shulian00/pdm-bearing> (accessed on 17 September 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Notation

$\{y_1, y_2, \dots, y_n\}$	RUL classification results
N	The number of random data points
$\mu\sigma$	The expectation and standard deviation of the truncated normal distribution
t_s	Prediction cycle duration
$\{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\}$	Ranges of RUL categories when calculating the distribution
$\{I_1, I_2, \dots, I_n\}$	Ideal expectation values corresponding to RUL categories
$\{t_1, t_2, \dots, t_m\}$	Time points at which the RUL probability density value needs to be obtained
C_p, t_p	Preventive maintenance cost and time required
C_c, t_c	Corrective replacement cost and time required
C_d, Et_d	Downtime cost per unit of time and estimated downtime
t	Maintenance time
Δt	The time interval used to determine whether shutdown is required for maintenance
t_0	Run time in the current maintenance cycle
Et_{total}	Estimated duration of the maintenance cycle
$P_i(t), f(x)$	Cumulative distribution value and probability density value of RUL
$EC, EC_{\text{max}}, EC_{\text{min}}$	The expectation of maintenance cost rate and its upper and lower limits
EA, EA_{min}	The expectation of availability and its lower limit
ER, ER_{min}	The expectation of reliability and its lower limit
Z_C, Z_A, Z_R	Expectations of maintenance cost rate, availability, and reliability after convergence and normalization
Z^+, Z^-	The optimal and worst solutions of the TOPSIS method
D^+, D^-	Euclidean distances between the three normalized indexes and the optimal and worst solutions
$\omega_C, \omega_A, \omega_R$	Weights of normalized maintenance cost rate, availability, and reliability
C_D	Comprehensive optimization target
L	The actual useful life of equipment or components in the current maintenance cycle
$EC_{\text{DR}}, EC_{\text{DN}}$	Cost rate expectations for immediate maintenance and no maintenance temporarily
δ	A sign indicating whether to shut down for maintenance
$t_{\text{PeM}}, t_{\text{CPdM}}, t_{\text{DPdM}}$	Maintenance time under PeM, C-PdM, and D-PdM policies

References

1. Nguyen, K.T.P.; Medjaher, K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliab. Eng. Syst. Saf.* **2019**, *188*, 251–262. [[CrossRef](#)]
2. Serradilla, O.; Zugasti, E.; De Okariz, J.R.; Rodriguez, J.; Zurutuza, U. Methodology for data-driven predictive maintenance models design, development and implementation on manufacturing guided by domain knowledge. *Int. J. Comput. Integr. Manuf.* **2022**, *35*, 1310–1334. [[CrossRef](#)]
3. Xu, X. Machine Tool 4.0 for the new era of manufacturing. *Int. J. Adv. Manuf. Technol.* **2017**, *92*, 1893–1900. [[CrossRef](#)]
4. Shaheen, B.; Kocsis, Á.; Németh, I. Data-driven failure prediction and RUL estimation of mechanical components using accumulative artificial neural networks. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105749. [[CrossRef](#)]
5. Dalzochio, J.; Kunst, R.; Barbosa, J.L.V.; Vianna, H.D.; De Oliveira Ramos, G.; Pignaton, E.; Binotto, A.; Favilla, J. ELFpm: A machine learning framework for industrial machines prediction of remaining useful life. *Neurocomputing* **2022**, *512*, 420–442. [[CrossRef](#)]
6. Gupta, M.; Wadhvani, R.; Rasool, A. A real-time adaptive model for bearing fault classification and remaining useful life estimation using deep neural network. *Knowl.-Based Syst.* **2023**, *259*, 110070. [[CrossRef](#)]
7. Tseng, F.; Filev, D.; Yildirim, M.; Chinnam, R.B. Online System Prognostics with Ensemble Models and Evolving Clustering. *Machines* **2022**, *11*, 40. [[CrossRef](#)]
8. Mosallam, A.; Medjaher, K.; Zerhouni, N. Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. *J. Intell. Manuf.* **2016**, *27*, 1037–1048. [[CrossRef](#)]
9. Behera, S.; Misra, R. A multi-model data-fusion based deep transfer learning for improved remaining useful life estimation for IIOT based systems. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105712. [[CrossRef](#)]
10. Akrim, A.; Gogu, C.; Vingerhoeds, R.; Salaün, M. Self-Supervised Learning for data scarcity in a fatigue damage prognostic problem. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105837. [[CrossRef](#)]
11. Nguyen, K.T.P.; Medjaher, K.; Tran, D.T. A review of artificial intelligence methods for engineering prognostics and health management with implementation guidelines. *Artif. Intell. Rev.* **2022**, *56*, 3659–3709. [[CrossRef](#)]
12. Custode, L.L.; Mo, H.; Ferigo, A.; Iacca, G. Evolutionary Optimization of Spiking Neural P Systems for Remaining Useful Life Prediction. *Algorithms* **2022**, *15*, 98. [[CrossRef](#)]
13. Alaswad, S.; Xiang, Y. A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliab. Eng. Syst. Saf.* **2017**, *157*, 54–63. [[CrossRef](#)]
14. Borrero, J.S.; Akhavan-Tabatabaei, R. Time and inventory dependent optimal maintenance policies for single machine workstations: An MDP approach. *Eur. J. Oper. Res.* **2013**, *228*, 545–555. [[CrossRef](#)]
15. Papakonstantinou, K.G.; Shinozuka, M. Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory. *Reliab. Eng. Syst. Saf.* **2014**, *130*, 202–213. [[CrossRef](#)]
16. Papakonstantinou, K.G.; Shinozuka, M. Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation. *Reliab. Eng. Syst. Saf.* **2014**, *130*, 214–224. [[CrossRef](#)]
17. Mosayebi Omshi, E.; Grall, A.; Shemehsavar, S. A dynamic auto-adaptive predictive maintenance policy for degradation with unknown parameters. *Eur. J. Oper. Res.* **2020**, *282*, 81–92. [[CrossRef](#)]
18. Oakley, J.L.; Wilson, K.J.; Philipson, P. A condition-based maintenance policy for continuously monitored multi-component systems with economic and stochastic dependence. *Reliab. Eng. Syst. Saf.* **2022**, *222*, 108321. [[CrossRef](#)]
19. Cai, H.; Jia, X.; Feng, J.; Li, W.; Pahren, L.; Lee, J. A similarity based methodology for machine prognostics by using kernel two sample test. *ISA Trans* **2020**, *103*, 112–121. [[CrossRef](#)]
20. Zhao, Z.; Wu, J.; Wong, D.; Sun, C.; Yan, R. Probabilistic remaining useful life prediction based on deep convolutional neural network. In Proceedings of the 9th International Conference on Through-Life Engineering Service, Cranfield, UK, 3–4 November 2020.
21. Caceres, J.; Gonzalez, D.; Zhou, T.T.; Droguett, E.L. A probabilistic Bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties. *Struct. Control Health Monit.* **2021**, *28*, e2811. [[CrossRef](#)]
22. Deutsch, J.; He, D. Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 11–20. [[CrossRef](#)]
23. Li, Z.; Wu, D.; Hu, C.; Terpenney, J. An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction. *Reliab. Eng. Syst. Saf.* **2019**, *184*, 110–122. [[CrossRef](#)]
24. Lima, A.L.D.C.D.; Aranha, V.M.; Nascimento, E.G.S. Predictive maintenance applied to mission critical supercomputing environments: Remaining useful life estimation of a hydraulic cooling system using deep learning. *J. Supercomput.* **2022**, *79*, 4660–4684. [[CrossRef](#)]
25. Akkad, K.; He, D. A dynamic mode decomposition based deep learning technique for prognostics. *J. Intell. Manuf.* **2023**, *34*, 2207–2224. [[CrossRef](#)]
26. Khooran, M.; Golbahar Haghighi, M.R.; Malekzadeh, P. Remaining Useful Life Prediction by Stacking Multiple Windows Networks with a Ridge Regression. *Iran. J. Sci. Technol. Trans. Mech. Eng.* **2022**, *47*, 583–594. [[CrossRef](#)]
27. Forouzandeh Shahraki, A.; Al-Dahidi, S.; Rahim Taleqani, A.; Yadav, O.P. Using LSTM neural network to predict remaining useful life of electrolytic capacitors in dynamic operating conditions. *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.* **2022**, *237*, 16–28. [[CrossRef](#)]

28. Listou Ellefsen, A.; Bjørlykhaug, E.; Æsøy, V.; Ushakov, S.; Zhang, H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliab. Eng. Syst. Saf.* **2019**, *183*, 240–251. [[CrossRef](#)]
29. Hesabi, H.; Nourelfath, M.; Hajji, A. A deep learning predictive model for selective maintenance optimization. *Reliab. Eng. Syst. Saf.* **2022**, *219*, 108191. [[CrossRef](#)]
30. Abdelghafar, S.; Khater, A.; Wagdy, A.; Darwish, A.; Hassanien, A.E. Aero engines remaining useful life prediction based on enhanced adaptive guided differential evolution. *Evol. Intell.* **2022**. [[CrossRef](#)]
31. Soualhi, M.; Nguyen, K.T.P.; Medjaher, K.; Nejari, F.; Puig, V.; Blesa, J.; Quevedo, J.; Marlasca, F. Dealing with prognostics uncertainties: Combination of direct and recursive remaining useful life estimations. *Comput. Ind.* **2023**, *144*, 103766. [[CrossRef](#)]
32. Shutin, D.; Bondarenko, M.; Polyakov, R.; Stebakov, I.; Savin, L. Method for On-Line Remaining Useful Life and Wear Prediction for Adjustable Journal Bearings Utilizing a Combination of Physics-Based and Data-Driven Models: A Numerical Investigation. *Lubricants* **2023**, *11*, 33. [[CrossRef](#)]
33. An, D. Prediction-Interval-Based Credibility Criteria of Prognostics Results for Practical Use. *Processes* **2022**, *10*, 473. [[CrossRef](#)]
34. Baur, M.; Albertelli, P.; Monno, M. A review of prognostics and health management of machine tools. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 2843–2863. [[CrossRef](#)]
35. Bracale, A.; De Falco, P.; Di Noia, L.P.; Rizzo, R. Probabilistic State of Health and Remaining Useful Life Prediction for Li-ion Batteries. In Proceedings of the 2021 IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 2–5 February 2021.
36. Li, G.Y.; Yang, L.; Lee, C.G.; Wang, X.H.; Rong, M.Z. A Bayesian Deep Learning RUL Framework Integrating Epistemic and Aleatoric Uncertainties. *IEEE Trans. Ind. Electron.* **2021**, *68*, 8829–8841. [[CrossRef](#)]
37. Gao, G.; Que, Z.; Xu, Z. Predicting Remaining Useful Life with Uncertainty Using Recurrent Neural Process. In Proceedings of the 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, 11–14 December 2020.
38. Li, X.C.; Mba, D.; Lin, T.R.; Yang, Y.J.; Loukopoulos, P. Just-in-time learning based probabilistic gradient boosting tree for valve failure prognostics. *Mech. Syst. Sig. Process.* **2021**, *150*, 107253. [[CrossRef](#)]
39. Tamssaouet, F.; Nguyen, K.T.P.; Medjaher, K.; Orchard, M. A Fresh New Look on System-level Prognostic: Handling Multi-component Interactions, Mission Profile Impacts, and Uncertainty Quantification. *Int. J. Progn. Health Manag.* **2021**, *12*, 1–15. [[CrossRef](#)]
40. Nguyen, K.T.P.; Medjaher, K.; Gogu, C. Probabilistic deep learning methodology for uncertainty quantification of remaining useful lifetime of multi-component systems. *Reliab. Eng. Syst. Saf.* **2022**, *222*, 108383. [[CrossRef](#)]
41. Huynh, K.T. An adaptive predictive maintenance model for repairable deteriorating systems using inverse Gaussian degradation process. *Reliab. Eng. Syst. Saf.* **2021**, *213*, 107695. [[CrossRef](#)]
42. Thoppil, N.M.; Vasu, V.; Rao, C.S.P. Bayesian Optimization LSTM/bi-LSTM Network With Self-Optimized Structure and Hyperparameters for Remaining Useful Life Estimation of Lathe Spindle Unit. *J. Comput. Inf. Sci. Eng.* **2022**, *22*, 021012. [[CrossRef](#)]
43. Thoppil, N.M.; Vasu, V.; Rao, C.S.P. An Integrated Learning Algorithm for Vibration Feature Selection and Remaining Useful life Estimation of Lathe Spindle Unit. *J. Fail. Anal. Prev.* **2022**, *22*, 1693–1701. [[CrossRef](#)]
44. Pater, I.D.; Mitici, M. Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105582. [[CrossRef](#)]
45. Rathore, M.S.; Harsha, S.P. An attention-based stacked BiLSTM framework for predicting remaining useful life of rolling bearings. *Appl. Soft Comput.* **2022**, *131*, 109765. [[CrossRef](#)]
46. Shimazaki, H.; Shinomoto, S. Kernel bandwidth optimization in spike rate estimation. *J. Comput. Neurosci.* **2010**, *29*, 171–182. [[CrossRef](#)]
47. Ruschel, E.; Santos, E.a.P.; Loures, E.D.F.R. Industrial maintenance decision-making: A systematic literature review. *J. Manuf. Syst.* **2017**, *45*, 180–194. [[CrossRef](#)]
48. Senthil, C.; Sudhakara Pandian, R. Proactive Maintenance Model Using Reinforcement Learning Algorithm in Rubber Industry. *Processes* **2022**, *10*, 371. [[CrossRef](#)]
49. Raghav, Y.S.; Mradula, Varshney, R.; Modibbo, U.M.; Ahmadini, A.a.H.; Ali, I. Estimation and Optimization for System Availability Under Preventive Maintenance. *IEEE Access* **2022**, *10*, 94337–94353. [[CrossRef](#)]
50. Souza, M.L.H.; Costa, C.a.D.; Ramos, G.D.O.; Righi, R.D.R. A survey on decision-making based on system reliability in the context of Industry 4.0. *J. Manuf. Syst.* **2020**, *56*, 133–156. [[CrossRef](#)]
51. Gul, S.; Aydogdu, A. Novel distance and entropy definitions for linear Diophantine fuzzy sets and an extension of TOPSIS (LDF-TOPSIS). *Expert Syst.* **2023**, *40*, e13104. [[CrossRef](#)]
52. Lee, J.; Qiu, H.; Yu, G.; Lin, J. Rexnord Technical Services. In *NASA Ames Prognostics Data Repository*; NASA Ames Research Center: Silicon Valley, CA, USA, 2007.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.