International Journal of Industrial Engineering & Management Science, Vol. 8, Issue 2, (2022) 8-17

University of Hormozgan

International Journal of Industrial Engineering & Management Science

journal homepage: www.ijiems.com

Estimating the remaining useful life of equipment based on an optimal deep learning model and cross-correlation based similarity analysis

Amir Moballeghtohid[a,*], Fazilat Ahmadzade[b], Omid Ahmadzade[c]

[a] CMO at REIS Future Canada Inc, Winnipeg, Canada
[b] CEO at REIS Future Canada Inc, Winnipeg, Canada
[c] CTO at REIS Future Canada Inc, Winnipeg, Canada

A R T I C L E   I N F O

Article history
Received: 2022-06-16
Received in revised form: -
Accepted: 2022-11-20

Keywords:
Prognostics and health management
Remaining useful life
Deep learning
Long short-term memory
Cross-correlation

A B S T R A C T

Determining the remaining useful life (RUL) of key assets in a manufacturing company is one of the most important maintenance engineering activities to improve system reliability and reduce maintenance costs. Knowing the RUL of the equipment can help the decision-making process regarding the proper maintenance of the equipment (for example, repair or replacement). In this regard, one of the challenges is to determine the appropriate forecasting model. This includes designing a mathematical model as well as finding a model that is trained with the most similar data to the data obtained from the current state of the equipment. In this research, to design an appropriate forecasting model, a DE algorithm is proposed to optimize the LSTM deep learning model architecture. Also, to find a suitable reference forecasting model, the cross-correlation criterion has been used as a similarity index. This index takes into account time lags and can determine the most similar learning data set to the current state of the equipment data. To evaluate the proposed model, the FEMTO-ST Institute bearings data were used, which included run-to-failure vibration data of 6 learning bearings and 11 test bearings. To evaluate the proposed optimized forecasting model, competing forecasting models including optimized MLP, optimized SVR, and optimized GPR has been used. Also, the proposed similarity index (cross-correlation) has been compared with the Pearson correlation coefficient and inverse Euclidean distance. The evaluation results show that the proposed model of this research has a better performance than competing models.

1. Introduction

With the increasing level of competition among manufacturing companies, the importance and sensitivity of equipment health have also increased. Companies are making more efforts to minimize downtime on their product line. Due to this issue, the focus of maintenance researchers on CBM and PHM methods has increased. These methods are based on actions such as momentary monitoring of equipment, information extraction (such as information from vibration sensors, temperature, etc., or physical information such as cracking, wear, etc.), data analysis, and final decision on maintenance strategy [1]. Estimating the RUL, i.e. determining the time to equipment failure, is one of the most important steps in the PHM process. Knowing the RUL of the equipment will help you make the optimal decision to maintain, repair, or replace it [2].

In general, in PHM-related research, two general approaches to estimating the RUL have been proposed: the physics of failure approach and the data-driven approach [3]. In the physics of failure approach, the RUL is estimated using a physical model that relates failures (such as cracks, corrosion, wear, etc.) to the life of the equipment. One of the most important drawbacks of this method is determining an accurate mathematical model due to the high sensitivity of the model parameters and the difficulty of accurately identifying equipment failures and determining the extent of failure [4]. In the data-driven approach, the RUL is estimated based on the data extracted from sensors (such as vibration, temperature, pressure, etc.), and the analysis of this data is done using statistical methods and artificial intelligence [5].

So far, a lot of research has been done based on the mentioned approaches for assessing RUL. Variety in artificial intelligence, machine learning, and statistical methods for determining deterioration and forecasting the remaining time until failure has led to the development of various processes. As an example in reference [6], some types of these methods have been studied according to the type of equipment (bearings, shafts, gears, pumps, and cranes). The most important part of rotating machines is the bearing and 45-55%

* Corresponding author.
E-mail address: amobalegh@yahoo.com

of system failures are due to bearing failure [7]. Our focus in this paper is on forecasting the RUL based on a data-driven approach.

One of the challenges in estimating the remaining useful life is designing the estimation model. A proper estimation model can increase the accuracy of the forecast. Recently, models based on deep learning have shown their high capability in forecasting and the long short-term memory (LSTM) network is one of these successful models. The advantages of LSTM compared to the recurrent neural network (RNN) are considering long-term dependencies and also controlling the level of forgetfulness of this dependence. Despite the high advantages of the LSTM model, the architectural design of this model has a significant effect on its accuracy. Selecting the optimal hyperparameters and the correct activation functions that can be used can increase the accuracy of the model. Another challenge in estimating the remaining useful life is to determine a reference model for forecasting. Because learning data have different behaviors, the models fitted by each of these data can produce different results in predicting the RUL. Therefore, there is a need for a way to determine the appropriate reference model based on the similarity of learning and test data. According to the mentioned issues, in this research, a process for estimating the RUL for rolling bearings is presented in which the LSTM is used to predict the RUL. To optimize the LSTM, due to the nonlinearity of the optimization model, a meta-heuristic approach based on the differential evolution (DE) algorithm is proposed. Also, to determine the reference model for forecasting new data (data that we intend to use to estimate the remaining useful life), the cross-correlation method has been used to determine the similarity between learning data and new data.

The rest of the paper is organized as follows: In Section 2, the works related to determining the RUL of rolling element bearings based on deep learning are presented. In Section 3, the proposed method is described. In Section 4, the case study of the research is presented including data, evaluation method, and results and Section 5 concludes the paper.

2. Literature review

In recent years, much research has been done on the design of RUL estimation processes. Also, several review articles have been presented to collect and analyze these researches. For example, Ferreira & Gonçalves (2022) had an overview of RUL estimation models using machine learning [8]. Aloud & Alkhamees (2021) Focus on research related to RUL estimation of lithium batteries [9]. Sayyad et al. (2021) Reviewed articles related to the use of artificial intelligence in the design of RUL determination models [10]. In this section, we focus on articles related to the use of machine learning techniques in predicting bearing RUL .

Li et al. (2019) focus on The time-frequency domain information for prognostics, and multi-scale feature extraction using convolutional neural networks [11]. A data-driven framework is proposed by Cheng et al.(2020) to exploit the adoption of deep convolutional neural networks (CNNs) in predicting the RULs of bearings. raw vibrations of training bearings are first processed using the Hilbert–Huang transform to construct a novel nonlinear degradation energy indicator. The CNN is then employed to identify the hidden pattern between the extracted degradation energy indicator and the raw vibrations of training bearings. Shen et al. (2021) propose a physics-informed deep learning approach that consists of a simple threshold model and a deep convolutional neural network (CNN) model for bearing fault detection. In the proposed physics-informed deep learning approach, the threshold model first assesses the health classes of bearings based on the known physics of bearing faults. Then, the CNN model automatically extracts high-level characteristic features from the input data and makes full use of these features to predict the health class of a bearing. Dong et al. (2021) use transfer learning and deep hierarchical feature extraction to extract the features of vibration. degradation assessment is transformed to the classification task of degradation pattern, which divides the degradation process into normal, slight fault, fault development, and damage patterns. The hierarchical network with random weight parameters is introduced to extract the local sub-band characteristics of the spectrum, in which the multiple alternately convolution and pooling layers without supervised fine-tuning are employed [12]. Zhang et al. (2021) propose a transfer learning method for remaining useful life predictions using deep representation regularization. The practical and challenging scenario is investigated, where the training and testing data are from different machinery operating conditions, and no target-domain run-to-failure data is available for training [13]. Ding et al. (2021) present a deep convolutional neural network (DCNN) model without a pooling layer, which consists of three convolutional layers and two fully connected layers to estimate the RUL [14]. Huang et al. (2021) introduce the new approach using transfer depth-wise separable convolution recurrent network (TDSCRN) for RUL estimation of bearing [15]. Su et al. (2021) present a two-stage process for estimating the RUL: in the first stage, a feature pre-extraction mechanism is designed to pre-extract the low-level features in relatively high dimensional space, which requires no additional manual operations of feature fusion and feature selection in existing methods. In the second stage, an adaptive transformer, a new deep model integrating the attention mechanism and the recurrent architecture, is proposed to model the relationships between these low-level features and the RULs directly, which suppresses the issue of vanishing gradients and is more suitable for representing the complex temporal degradation characteristics. Ding et al. (2021) introduce a method called deep transfer metric learning for kernel regression (DTMLKR) and applied it to the RUL prediction of bearings under multiple operating conditions. This method combines deep metric learning with transfer learning (TL) to solve regression problems [16]. Zhao & Yuan (2021) proposes a scheme that contains both classification and regression, where the 2D-DCNN based classifier and predictors are built concerning typical fault conditions of a bearing. For the online prediction, the raw signals are spanned in the time-frequency domain and then transferred into images as the input of the scheme. The classifier is used to monitor the vibration of rolling bearings for online fault recognition and excite the corresponding predictor for RUL prediction once a fault is detected. The output from the predictor is amended by the proposed adaptive delay correction method as the final prediction results [17]. Chen et al. (2022) have used a new multi-scale long-term recurrent convolutional network framework with wide first layer kernels and residual shrinkage building unit (MSWR-LRCN) to estimate the remaining bearing life [18].The major difference from the previous deep neural network is that this new network organically combines the attention mechanism with a multi-scale feature fusion strategy, and improves the anti-noise ability of the entire network. Hu et al. (2022) propose a novel method called Deep Feature Disentanglement

Transfer Learning Network (DFDTLN) to extract domain-invariant features. In the proposed method, shared domain-invariant representations and private representations are disentangled by a pair of joint learning autoencoders. The effectiveness of the proposed method is verified using IEEE PHM Challenge 2012 dataset. The comparison results show the deep features extracted by DFDTLN are more domain-invariant and suitable for RUL prediction [19]. Wan et al. (2022) present a novel deep learning framework with multi-branch networks, which are called convolutional long short-term memory fusion networks (CLSTMF) for RUL prediction with multi-sensor data. In each branch network, shallow features of a single sensor's data are extracted by the convolutional layer of the convolutional neural network (CNN), and then the convolutional long short-term memory (CLSTM) network is employed to capture deep temporal features from these shallow features. Meanwhile, a novel information transfer layer (ITL) is developed to fuse the multi-sensor data's features captured with CLSTM in different branch networks [20]. The use of contrastive learning to maintain mutual information may introduce unstable negative samples. To overcome these issues Zhuang et al. (2022) present a metric adversarial domain adaptation approach (MADA) proposed to evaluate the bearing RULs under multiple working conditions. More specifically, an adversarial domain adaptation architecture with a supervised positive contrastive module is developed to consider mutual information without a negative sample, further learning domain invariant features. Also, the dual self-attention module is designed to extract multi-scale contextual semantics between degradation features [21]. Sparse representation is a practical approach for mining fault information from vibration signals. Zhou et al. (2022) proposes a novel end-to-end deep network-based sparse denoising (DNSD) framework based on a model-data-collaborative linkage framework for RUL estimation of bearing [22]. Table 1 provides a summary of the mentioned researches.

Table 1: Recent research related to RUL Bearing estimation based on deep learning approaches

References	Method
[1]	multi-scale feature extraction is implemented using convolutional neural networks
[2]	Hilbert–Huang transform based feature extraction and CNN for forecasting
[3]	a physics-informed deep learning approach that consists of a simple threshold model and a deep convolutional neural network (CNN) model
[4]	transfer learning and deep hierarchical features extraction
[5]	transfer learning method for remaining useful life predictions using deep representation regularization
[6]	convolutional neural network
[7]	transfer depth-wise separable convolution recurrent network (TDSCRN)
[8]	adaptive transformer, a new deep model integrating the attention mechanism and the recurrent architecture, is proposed to model the relationships between these low-level features and the RULs directly
[9]	deep transfer metric learning for kernel regression (DTMLKR)
[10]	2D-DCNN based classifier
[11]	end-to-end deep network-based sparse denoising (DNSD) framework
[12]	adversarial domain adaptation architecture with a supervised positive contrastive module
[13]	multi-scale long-term recurrent convolutional network with wide first layer kernels and residual shrinkage building unit (MSWR-LRCN)
[14]	Deep Feature Disentanglement Transfer Learning Network (DFDTLN) to extract domain-invariant features
[15]	convolutional long short-term memory fusion networks (CLSTMF)
Proposed model	**Optimized deep LSTM model selected based on the cross-correlation similarity index**

3. Proposed Model

3.1. Methods

Long short-term memory

LSTM is an artificial neural network used in the fields of artificial intelligence and deep learning. LSTM as an extension of RNN has a strong capability of forecasting time series data [26]. The main difference between an RNN and LSTM is that LSTM can store long-range time dependency information and can suitably map between input and output data [26]. The LSTM network structure differs from the conventional perceptron architecture as it contains a cell and gates which controls the flow of information. Specifically, the LSTM contains an input gate, a forget gate, an internal state (cell memory), and an output gate as illustrated in Figure 1. The notations of Figure 1 are as follows [27]:

– $x(t_i)$: The input value

– $h(t_{i-1})$ and $h(t_i)$: The output value at time t_{i-1} and t_i.

– $c(t_{i-1})$ and $c(t_i)$: Cell states at time t_{i-1} and t_i.

– $b = \{b_a, b_f, b_c, b_o\}$ are biases of input gate, forget gate, internal state, and output gate.

– $\overrightarrow{W_1} = \{w_a, w_f, w_c, w_o\}$ are weight matrixes of input gate, forget gate, internal state, and output gate.

– $\overrightarrow{W_2} = \{w_{ha}, w_{hf}, w_{hc}, w_{ho}\}$ are the recurrent weights.

– $\vec{a} = \{a(t_i), f(t_i), c(t_i), o(t_i)\}$ are the output results for the input gate, forget gate, internal state, and output gate.

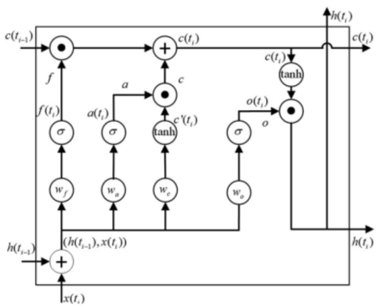

Figure 1: The structure of LSTM [27]

Considering these notations, the operation of LSTM is as follows: The forget gate $f(t_i)$ uses $x(t_i)$ and $h(t_{i-1})$ as input to compute the information to be preserved in $c(t_{i-1})$ using a transfer function. The input gate $a(t_i)$ takes $x(t_i)$ and $h(t_{i-1})$ to compute the value of $c(t_i)$. The output gate $o(t_i)$ performs regulation on the output of an LSTM cell by considering $c(t_i)$ and applying two transfer functions. Mathematically the forward learning of an LSTM is as follows:

$$a(t_i) = \sigma_g(w_a x(t_i) + w_{ha} h(t_{i-1}) + b_a) \quad (1)$$

$$f(t_i) = \sigma_g\left(w_f x(t_i) + w_{hf} h(t_{i-1}) + b_f\right) \quad (2)$$

$$c(t_i) = f_t\, c(t_{i-1}) + a_t\, \sigma_c\left(w_c x(t_i) + w_{hc}(h(t_{i-1}) + b_c)\right) \quad (3)$$

$$o(t_i) = \sigma_g(w_o x(t_i) + w_{ho} h(t_{i-1}) + b_o) \quad (4)$$

$$h(t_i) = o(t_i)\, \sigma_c\left(c(t_i)\right) \quad (5)$$

Where σ_g and σ_c are activation functions. Overall, the LSTM learns using the following steps:

(1) Compute the LSTM output using Eqs. (1)–(5) (forward learning).
(2) Compute the error between the resulted data and input data of each layer.
(3) The error is reversely propagated to the input gate, cell, and forget gate.
(4) Based on the error term, the weight of each gate is updated using an optimization algorithm.

The above four-step process is repeated for a given number of iterations and the optimal values of weights and biases are obtained.

Cross-correlation

Cross-correlation is a measure of similarity of two series as a function of the displacement of one relative to the other. The true cross-correlation sequence of two jointly stationary random processes, x_n and y_n with length N is given by [28]:

$$\hat{R}_{xy}(m) = E\{x_{n+m} y_n^*\} = E\{x_n y_{n-m}^*\} = \begin{cases} \sum_{n=0}^{N-m-1} x_{n+m} y_n^* & m \geq 0 \\ \hat{R}_{xy}^*(-m) & m < 0 \end{cases} \quad (6)$$

Where $\hat{R}_{xy}(m)$ is the estimation of cross-correlation in lag m, the asterisk $(*)$ denotes complex conjugation, and E is the expected value operator.

Differential evolution

Differential Evolution (DE) is an evolutionary, stochastic, population-based optimization algorithm introduced by Storn and Price in 1996 [16]. **Error! Reference source not found.** depicts the flowchart of DE. In DE an optimal solution is explored from a randomly generated starting population using three evolutionary operations: mutation, crossover, and selection. For each generation, the individuals of the current population become target vectors and their fitness function value is calculated. Then the mutation operation produces a mutant vector m_{ij} for each individual target vector x_j using the Equation (7):

$$m_{ij} = x_{r_1 j} + F \times \left(x_{r_2 j} - x_{r_3 j}\right) \quad (7)$$

Where $r_1 \neq r_2 \neq r_3$ are random and mutually exclusive integers. The mutant vector along with the target vector is further passed through the crossover operation to produce a trial vector as follows:

$$t_{ij} = m_{ij}\ if\ rand[0,1] \ \leq cr\ or\ j = j_{rand}\ else\ x_{ij} \quad (8)$$

The trial vector replaces the target vector if its fitness value is better than the target vector. So it can be summarized that mutation enlarges the search space, Crossover recapitulates previously successful individuals and selection encourages the survival of the fittest. The mutation, crossover and selection operations are repeated until some termination condition is reached.

3.2. Remaining useful life estimation based on optimal deep learning and similarity analysis

Error! Reference source not found. shows the proposed RUL estimation process. There are two datasets of learning and testing in this process. The learning set includes N bearings and the test set includes M bearings. Learning set bearings were used to fit the forecasting models and test set bearings were used to evaluate the proposed process. In the following, the proposed process is described in terms of its general steps:

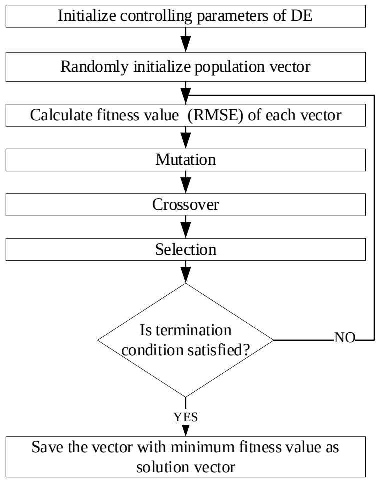

Figure 2: The flowchart of DE [16]

Feature extraction

In most cases, raw data such as vibration, temperature, pressure, etc., can't reflect all the characteristics related to the equipment failure process. For this reason, features are extracted from raw data to increase the ability to describe the failure process. This study focuses on the features of bearing vibrations including entropy, energy, RMS, skewness, and kurtosis. These features have been considered by researchers in many studies to estimate the remaining useful life. One of the drawbacks of these features is their low trendability and smoothness, which confuses the forecasting model. For this reason [17] has used the cumulative values of these features, which has led to higher accuracy of the forecasting model. In this process, we also use the cumulative values of the entropy, energy, RMS, skewness, and kurtosis features of the vibration.

Optimization of LSTM

In **Error! Reference source not found.**, the proposed LSTM model for RUL forecasting is presented, which is defined as a regression model. As can be seen, the inputs of the LSTM are cumulative features and the output is RUL. Layers include the input layer, LSTM layer, and fully connected layer. In this step, the optimal LSTM models for each of the learning set bearings are trained and stored. There are settings that can affect the accuracy of the LSTM and should be optimized:

- The number of hidden units: The number of hidden units corresponds to the amount of information remembered between time steps (the hidden state). The hidden state can contain information from all previous time steps, regardless of the sequence length. If the number of hidden units is too large, then the layer might overfit to the training data. This value can vary from a few dozen to a few thousand.
- Size of mini-batch: A mini-batch is a subset of the training set that is used to evaluate the gradient of the loss function and update the weights

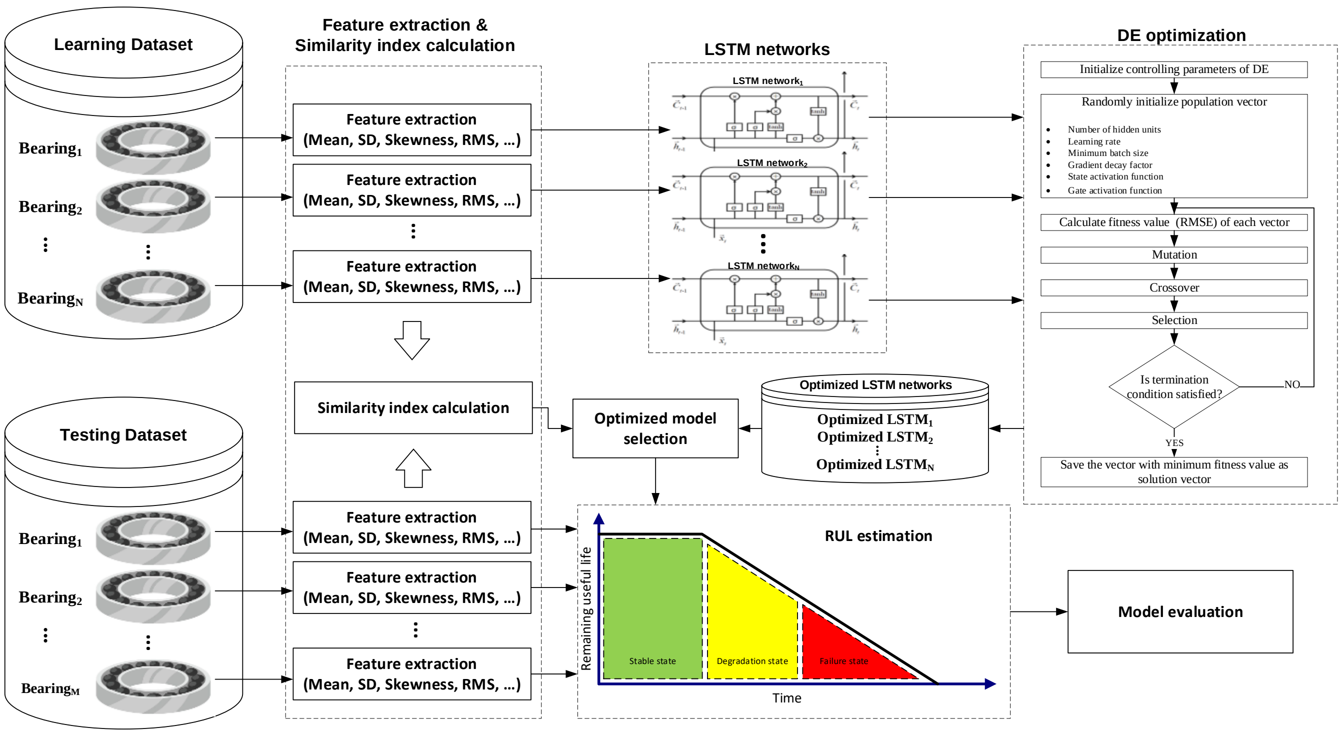

Figure 3: Proposed RUL estimation process

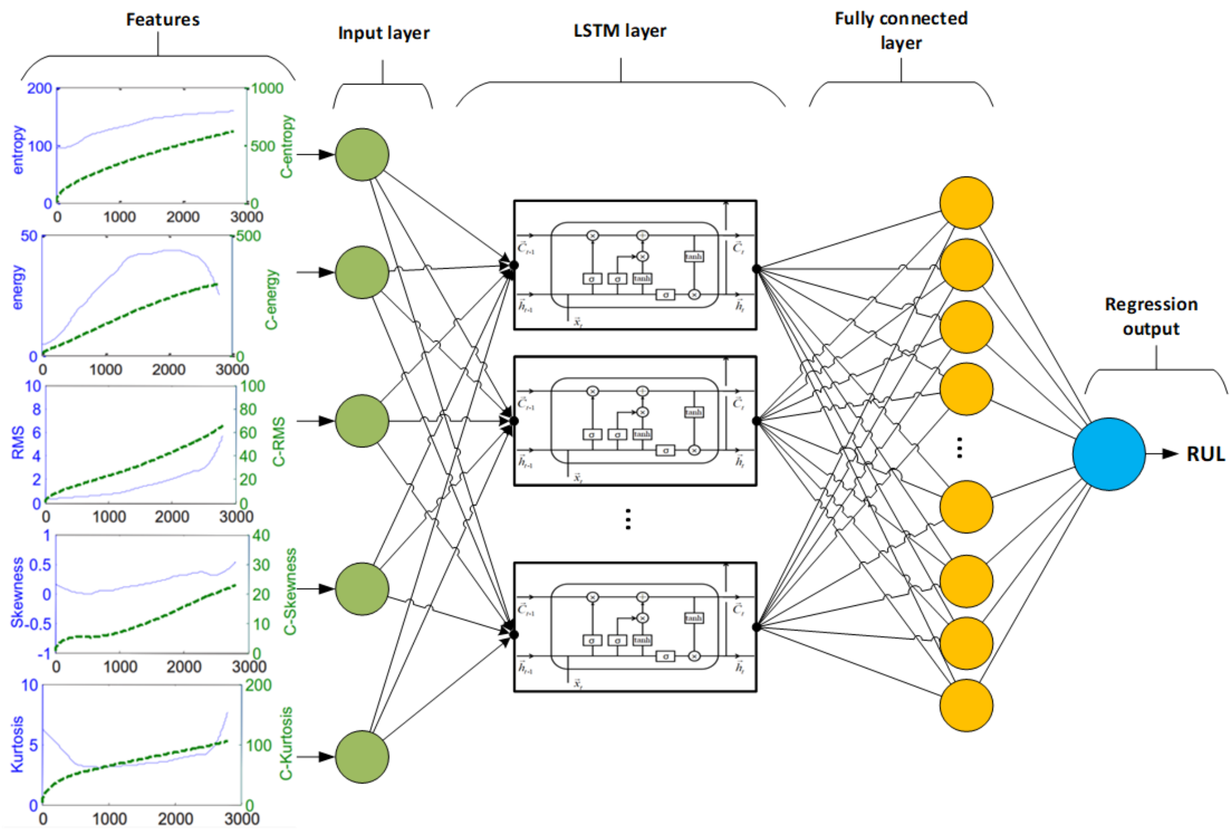

Figure 4: Proposed LSTM model architecture

- Learning rate: The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. if the learning rate is too low, then training can take a long time. If the learning rate is too high, then training might reach a suboptimal result or diverge.
- The maximum number of epochs: An iteration is one step taken in the gradient descent algorithm towards minimizing the loss function using a mini-batch. An epoch is the full pass of the training algorithm over the entire training set.
- State Activation function : Activation function to update the cell and hidden state ($\sigma_c = tanh$ or $\sigma_c =$ softsign function $= \frac{x}{1+|x|}$)
- Gate activation function: Activation function to apply to the gates

$$(\sigma_g = \begin{cases} 0 & if\ x < -2.5 \\ 0.2x + 0.5 & if -2.5 < x < 2.5 \\ 1 & if\ x > 2.5 \end{cases}$$

In this research, the focus is on optimizing the mentioned settings. For this purpose, a differential evolution model is proposed to solve the LSTM optimization model. **Error! Reference source not found.** shows a chromosome of the differential evolution model and the range of each of the variables. The purpose of the optimization model is to minimize the estimation error or RMSE.

nhu	mbs	lr	mne	saf	gaf
[10,1000]	[5,1000]	[0.001,1]	[5,100]	{0,1}	{0,1}

Figure 5: Chromosome model of the proposed differential evolution algorithm - nhu: number of hidden units, mbs: minimum batch size, lr: learning rate, mne: maximum number of epochs, saf: state activation function ($0 = tanh, 1 = softsign$), gaf: gate activation function ($sigmoid\ function\ or\ hard\ sigmoid\ function$)

In **Error! Reference source not found.**, the pseudocode of the differential evolution model for optimizing the LSTM network for each of the learning bearings is presented:

```
1   for i ∈ {1,2,…,N = number of training dataset berings}
2     Calculate Bearing_i features for each time step (X)
3     Prepare RUL for each time step (Y)
4     Initialize the controlling parameters of DE
5     Initialize the population vector (nhu, mbs, lr, mne, saf, gaf)
6     Fit the LSTM network with population vector setting (input: X, output: Y)
7     Calculate the fitness function (RMSE) for population
8     Mutation
9     Crossover
10    selection
11    Is termination condition satisfied? (Yes: go to next line, NO: go to 6)
12    Save the best trained LSTM_i network
13  End
```

Figure 6: Pseudocode of the differential evolution model for optimizing the LSTM network

The mutation operator for the variables nhu, mbs, lr, and mne in x_i which leads to the mutant solution V_i is considered as an Equation (9). For the variables saf and gaf, one of the values 0 or 1 is randomly assigned:

$$V_i = x_{r_1} + F.\left(x_{r_2} - x_{r_3}\right) \tag{9}$$

Where r_1, r_2, r_3, r_4 and r_5 are randomly generated exclusive integers within $[1, M]$. The scaling factor F is a positive control parameter to the scale difference vector. Each pair of target vectors x_i and their corresponding mutation vectors V_i are crossed to generate a test vector $U_i = (u_1, u_2, \dots, u_i)$. In the DE algorithm, a binomial crossover is defined as follows:

$$u_i = \begin{cases} v_i & if\ \left(rand_j(0,1) \leq CR\right)\ or\ \left(j = j_{rand}, j = 1,2,3,\dots,D\right) \\ x_i & otherwise \end{cases} \tag{10}$$

Where the crossover rate CR is a specified constant on $[0,1]$ which is used to control the duplicated proportion from the mutation vector. j_{rand} is a randomly selected integer on $[1, D]$. The optimization algorithm stops when the value of the objective function is not changed in 150 iterations.

Determining the appropriate forecasting model based on the cross-correlation similarity index

To forecast the RUL of new data, the appropriate model must be selected from the models trained by the learning data. Naturally, a suitable model is trained for learning data similar to new data. In this study, cross-correlation has been used to determine the level of similarity between time series. For this purpose, the similarity between the new ith data and the learning data is calculated as Equation (11):

$$similarity(i,j) = max\left\{R_{ij}(m), m \in [-l, l]\right\} \tag{11}$$

Where $R_{ij}(m)$ is the cross-correlation between the data of the ith testing bearing and the jth learning data in the lag m, and l is the maximum number of lags considered for the cross-correlation. After determining the similarity index of the ith bearing with all learning bearings, the learning data j^* with the largest similarity index $\left(j^* = \max_j similarity(i,j)\right)$ as the reference bearing and the model $LSTM_{j^*}$ used as a reference model for forecasting.

RUL estimation

After determining the appropriate model for each of the test data, the RUL is estimated using the selected model and the new data:

$$\widehat{RUL}_i = LSTM_{j^*}(features_i) \tag{12}$$

4. Case study

4.1. Data

PRONOSTIA (**Error! Reference source not found.**) is a laboratory platform designed to test and verify bearing failure detection and fault detection and prediction approaches [18]. This platform is designed in the AS2M department of FEMTO-ST Institute. The main purpose of PRONOSTIA is to provide real-life data that describes the bearing decay process over its entire life, i.e. until its total deterioration.

The PRONOSTIA platform makes it possible to perform run-to-failure tests. To avoid propagating the fault to the entire platform (and for safety reasons), it stops when the amplitude of the signal vibration exceeds 20g. **Error! Reference source not found.** (center) shows an example of what happens to bearings before and after failure, as well as the raw vibration signal collected throughout the test (right). Note that deteriorated bearings exhibit different behaviors and therefore lead to different test periods

Figure 7: PRONOSTIA platform (left), before and after bearing failure (center), bearing vibration signal (right) [31]

(up to failure). The data set collected by this system is generated under three conditions (Table 2):

- First operating conditions: 1800 rpm and 4000N
- Second operating conditions: 1650rpm and 4200N
- Third operating conditions: 1500rpm and 5000N

6 run-to-failure datasets are provided to create prediction models (learning datasets) and RUL estimates are required for the other 11 bearings. Vibration signals are collected for all test components.

There is no assumption for the type of damage that occurred (no information about the root and origin of the deterioration: bullets, inner ring, outer ring, cage ...)

Table 2: Overview of data generated by the PRONOSTIA platform

Dataset	Operating Condition		
	First operating condition	Second operating condition	Third operating condition
Learning dataset	Bearing 1-1	Bearing 2-1	Bearing 3-1
	Bearing 1-2	Bearing 2-2	Bearing 3-2
Testing dataset	Bearing 1-3	Bearing 2-3	Bearing 3-3
	Bearing 1-4	Bearing 2-4	
	Bearing 1-5	Bearing 2-5	
	Bearing 1-6	Bearing 2-6	
	Bearing 1-7	Bearing 2-7	

4.2. Results

Error! Reference source not found. shows the features and their cumulative values for the bearing 1-1. As can be seen, non-cumulative features are highly volatile, do not have a tangible trend, and are not smooth, therefore, they have low predictability. But as can be seen, cumulative features are quite trendy and also have much less volatility than non-cumulative features and therefore have high predictability.

After extracting the features of each learning bearing, the optimal LSTM network for each learning bearing must be trained. **Error! Reference source not found.** shows the trend change of the objective function (RMSE value for LSTM network) for each iteration of the DE algorithm. **Error! Reference source not found.** shows the optimal values for each of the optimization problem variables that represent the LSTM model settings.

Table 3: Optimized solutions of DE

Bearing name	nhu	mbs	lr	mne	saf	gaf
Bearing 1-1	101	160	0.007	41	0	0
Bearing 1-2	68	76	0.003	27	0	1
Bearing 2-1	42	59	0.009	14	1	1
Bearing 2-2	25	220	0.012	66	0	1
Bearing 3-1	170	95	0.029	72	1	0
Bearing 3-2	26	181	0.001	29	1	0

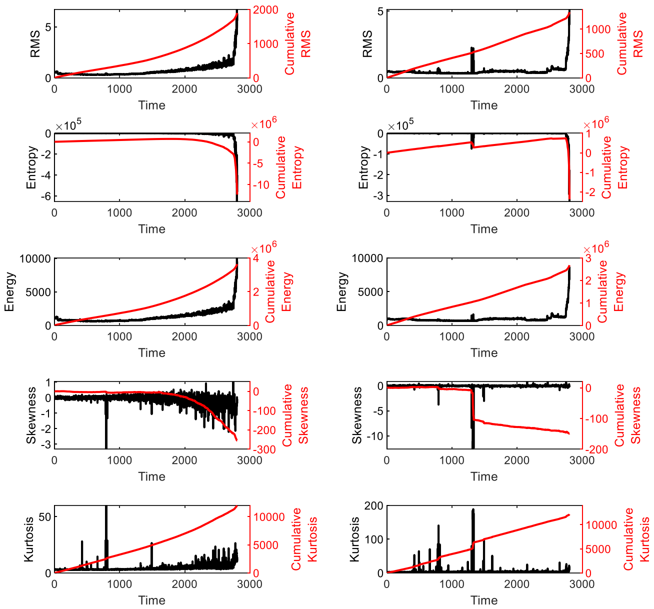

Figure 8: Features extracted from bearing 1-1 vibration data

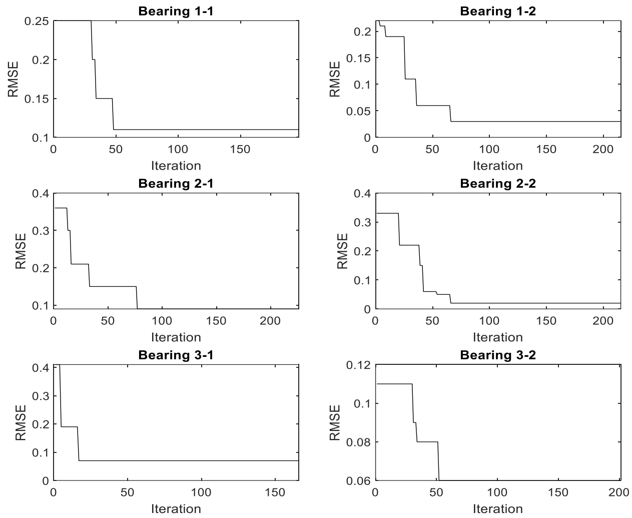

Figure 9: RMSE changes based on the DE iterations

After training the optimal LSTM model for each of the learning bearings, the similarity index for the test bearings is calculated. In **Error! Reference source not found.**, the similarity index, which is based on cross-correlation, is shown for each pair of learning and test bearings. This index is calculated based on horizontal, vertical, and average vibrations of these two values and the decision criterion is based on the average value. Accordingly, the reference LSTM network for each of the test bearings is shown in **Error! Reference source not found.**

Testing bearings	1-3 ↓	1-4 ↓	1-5 ↓	1-6 ↓	1-7 ↓	2-3 ↓	2-4 ↓	2-5 ↓	2-6 ↓	2-7 ↓	3-3 ↓
Optimized trained LSTM reference model	2-2	3-1	3-1	3-2	2-2	3-2	3-1	3-2	3-1	1-1	2-1

Figure 11: selected LSTM network based on similarity analysis

Finally, **Error! Reference source not found.** shows the estimated RUL, the actual RUL and percentage error of estimation

Comparison with competing models

To evaluate the proposed model, the results of the proposed model are compared with competing models. For this purpose, three prediction regression models have been used: multi-layer perceptron (MLP)[19], support vector regression (SVR) [20], and Gaussian process regression (GPR)[21]. These models, similar to the optimized LSTM model, are optimized using the DE algorithm. The optimized parameters for each of the models are:

- MLP: number of hidden layers, number of nodes in hidden layers, learning rate, activation function
- SVR: Kernel scale parameter, Number of iterations, Tolerance for gradient difference
- GPR: Initial value for the noise standard deviation of the Gaussian process model, Constant value of Sigma for the noise standard deviation of the Gaussian process model, Lower bound on the noise standard deviation.

Also, two similarity indices have been used to compare with cross-correlation: Pearson correlation coefficient and inverse Euclidean distance. The process of calculating RUL based on competing forecasting models and similarity indices is quite similar to the process proposed in this paper. After calculating the RUL, two evaluation criteria were used to compare the models. RMSE and SCORE criteria [22]:

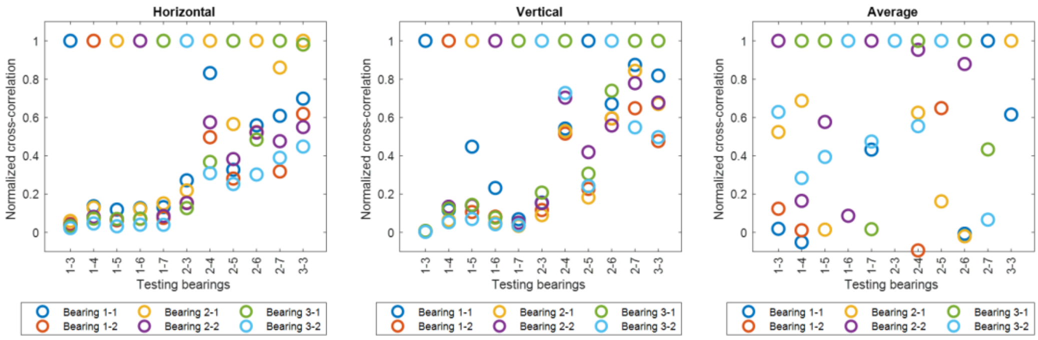

Figure 10: similarity index (normalized cross-correlation) for each pair of learning and test bearings

Table 4: Estimated and actual RUL for test bearings

Testing bearings	Bearing 1-3	Bearing 1-4	Bearing 1-5	Bearing 1-6	Bearing 1-7	Bearing 2-3	Bearing 2-4	Bearing 2-5	Bearing 2-6	Bearing 2-7	Bearing 3-3
Estimated RUL (seconds)	3020	121	1320	1270	8942	3211	1790	6734	410	650	690
Actual RUL (seconds)	5730	339	1610	1460	7570	7530	1390	3090	1290	580	820
Error (%)	47.29	64.30	18.01	13.01	-18.12	57.35	-28.77	-117.93	68.21	-12.06	15.85

$$ER_i = \frac{ActRUL_i - PredRUL_i}{ActRUL_i} \times 100 \quad (13)$$

$$A_i = \begin{cases} exp^{-\ln(0.5).\left(\frac{Er_i}{5}\right)} & if\ Er_i \le 0 \\ exp^{+\ln(0.5).\left(\frac{Er_i}{20}\right)} & if\ Er_i > 0 \end{cases} \quad (14)$$

$$SCORE = \frac{1}{N}\sum_{i=1}^{N} A_i \quad (15)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(ActRUL_i - PredRUL_i)^2}{N}} \quad (16)$$

Where ER_i is the percentage error, $ActRUL_i$ is the actual RUL of $bearing_i$, $PredRUL_i$ is the estimated RUL of $bearing_i$, N is the number of members of the test set. **Error! Reference source not found.** shows the RMSE and SCORE criteria for forecasting models and measurement indicators. As can be seen, the proposed model of this paper has shown better performance than other models. Also, the cross-correlation index has been able to find a better forecasting model. According to the SCORE criterion, after the proposed model, MLP, SVR, and GPR models are in order of performance, respectively. Also, after the cross-correlation index, Pearson and Euclidian indices were able to find the optimal estimation model, respectively. These results are also validated by considering the RMSE criterion.

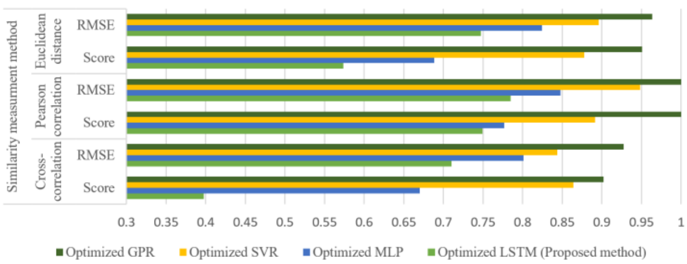

Figure 12: Comparison of forecasting models and similarity indices

5. Conclusion

Determining the RUL of equipment is one of the most important PHM activities. In this paper, a process for determining the RUL of a bearing is presented, which focuses on optimizing the forecasting model as well as finding a suitable trained forecasting reference model. For this purpose, a DE algorithm is proposed to optimize the LSTM deep learning model and its architectural design. The cross-correlation criterion was also used to find the reference model for forecasting. This criterion can determine the similarity of learning and test data by considering the co-movement with a time lag. To evaluate the proposed process, the FEMTO-ST Institute bearings dataset was used. The results obtained from the implementation of the proposed forecast model for the mentioned data have been compared with competing forecast models including MLP, SVR, and GPR models. Similar to the LSTM design process, competing forecasting models are optimized using the DE algorithm. The similarity indices used to compare with cross-correlation are Pearson correlation coefficient and inverse Euclidean distance. The results of this evaluation show that the proposed model (optimal LSTM model and cross-correlation) has better performance than competing models. The proposed research process can be used in situations where it is difficult to identify the forecast reference model due to a large number of learning data. In addition, using meta-heuristic models to optimize forecasting models can increase the possibility of finding the best forecasting model. The proposed research process can be developed in several ways. Utilization of autoregressive models such as LSTM autoregressive model or ARMA model for indirect prediction and model optimization can be used in future research. A comparison of other meta-heuristic models such as genetic algorithms to optimize prediction models can be considered as a research topic.

References

[1] Li, X., W. Zhang, and Q. Ding, *Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction.* Reliability engineering & system safety, 2019. **182**: p. 208-218.
[2] Cheng, C., et al., *A deep learning-based remaining useful life prediction approach for bearings.* IEEE/ASME Transactions on Mechatronics, 2020. **25**(3): p. 1243-1254.
[3] Shen, S., et al., *A physics-informed deep learning approach for bearing fault detection.* Engineering Applications of Artificial Intelligence, 2021. **103**: p. 104295.
[4] Dong, S., et al., *Transfer learning for bearing performance degradation assessment based on deep hierarchical features.* ISA transactions, 2021. **108**: p. 343-355.
[5] Zhang, W., et al., *Transfer learning using deep representation regularization in remaining useful life prediction across operating conditions.* Reliability Engineering & System Safety, 2021. **211**: p. 107556.
[6] Ding, H., et al., *A remaining useful life prediction method for bearing based on deep neural networks.* Measurement, 2021. **172**: p. 108878.
[7] Huang, G., Y. Zhang, and J. Ou, *Transfer remaining useful life estimation of bearing using depth-wise separable convolution recurrent network.* Measurement, 2021. **176**: p. 109090.
[8] Su, X., et al., *An end-to-end framework for remaining useful life prediction of rolling bearing based on feature pre-extraction mechanism and deep adaptive transformer model.* Computers & Industrial Engineering, 2021. **161**: p. 107531.
[9] Ding, Y., et al., *Remaining useful life estimation using deep metric transfer learning for kernel regression.* Reliability Engineering & System Safety, 2021. **212**: p. 107583.
[10] Zhao, B. and Q. Yuan, *A novel deep learning scheme for multi-condition remaining useful life prediction of rolling element bearings.* Journal of Manufacturing Systems, 2021. **61**: p. 450-460.
[11] Zhou, X., et al., *A hybrid denoising model using deep learning and sparse representation with application in bearing weak fault diagnosis.* Measurement, 2022. **189**: p. 110633.
[12] Zhuang, J., M. Jia, and X. Zhao, *An adversarial transfer network with supervised metric for remaining useful life prediction of rolling bearing under multiple working conditions.* Reliability Engineering & System Safety, 2022: p. 108599.
[13] Chen, Y., D. Zhang, and W.-a. Zhang, *MSWR-LRCN: A new deep learning approach to remaining useful life estimation of bearings.* Control Engineering Practice, 2022. **118**: p. 104969.
[14] Hu, T., et al., *Remaining useful life prediction of bearings under different working conditions using a deep feature disentanglement based transfer learning method.* Reliability Engineering & System Safety, 2022. **219**: p. 108265.
[15] Wan, S., et al., *Bearing remaining useful life prediction with convolutional long short-term memory fusion networks.* Reliability Engineering & System Safety, 2022. **224**: p. 108528.
[16] Storn, R. and K. Price, *Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.* Journal of global optimization, 1997. **11**(4): p. 341-359.
[17] Javed, K., et al. *Robust, reliable and applicable tool wear monitoring and prognostic: approach based on an improved-extreme learning machine*. in *Prognostics and Health Management (PHM), 2012 IEEE Conference on*. 2012. IEEE.
[18] Nectoux, P., et al. *PRONOSTIA: An experimental platform for bearings accelerated degradation tests*. in *IEEE International Conference on Prognostics and Health Management, PHM'12.* 2012. IEEE Catalog Number: CPF12PHM-CDR.
[19] Ramchoun, H., et al., *Multilayer perceptron: Architecture optimization and training.* 2016.
[20] Wu, C.-H., G.-H. Tzeng, and R.-H. Lin, *A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression.* Expert Systems with Applications, 2009. **36**(3): p. 4725-4735.
[21] Cao, Y., et al., *Efficient optimization for sparse Gaussian process regression.* Advances in Neural Information Processing Systems, 2013. **26**.
[22] Sutrisno, E., et al. *Estimation of remaining useful life of ball bearings using data driven methodologies*. in *Prognostics and Health Management (PHM), 2012 IEEE Conference on*. 2012. IEEE.