Bearing Remaining Useful Life Estimation Based on Encoder and Gated Recurrent Units

Dongdong Zhao, Liu Feng and Xiaodong Wang

***Abstract*—Bearings are to machinery like joints to humans, which means bearings are an integral part of the machinery. Recently, with the rapid development of sensors and artificial intelligence, a data-driven machine learning-based model for bearing remaining useful life (RUL) has become a powerful tool in academia and industry. However, there are also some existing problems, for example, the pre-set state-related threshold need to be manually determined, the bearing object being studied is under a single operating condition, RUL estimation in the entire life span in many studies is ignored and only a few samples as observed data to predicted and tested, which are all not conducive to the achievements transformation in actual engineering projects. In this paper, a novel generalized framework based on neural networks is proposed to solve these obstacles. Firstly, convolutional neural networks (CNN) and deep encoder (DE) are respectively used to extract features and reduce dimensions, obtaining dense and low dimensions health indicators (HIs). Secondly, Gated recurrent units (GRUs) are used to investigate time-series information from dense and low dimensional HIs. Thirdly, the proposed method was verified on two platforms and experimental results show that the proposed method is effective and general, and superior to other baselines.**

***Index Terms*—Bearing, Remaining useful life estimation, Convolutional neural network, Encoder, GRU**

I. Introduction

RECENTLY, prognostics health management (PHM) has become a research hotspot [1], [2], which includes fault detection, fault diagnosis [3], [4], state assessment and remaining useful life (RUL) estimation in machinery. Researches on PHM turns the post-failure repair into repair in advance, largely lower operating risks and costs, especially in the transportation industry such as aviation and rail transportation which usually have strong relationships with big assets. So, a large number of researchers in industry and academia also have paid their attention to PHM investigation in rotating machinery. In this paper, we mainly focus on the prediction of the RUL of bearing. From all current reports about RUL, the review [2] divided the RUL estimation model into three kinds: physics model-based methods [5], statistics model-based methods, and machine learning-based (ML-based) methods. The implementation in practical engineering of investigations on physical methods is difficult since in most scenarios, building a physical model for an object is not a simple task for there are many complex relationships with surround. However, we are not always supposed to set an assumption toward ideal scenes. The statistical model-based methods [6]–[13], also named empirical model-based approaches, severely rely on empirical knowledge to estimate the RUL. The ML-based approaches [14]–[21] are currently the most popular and efficient in RUL research, which take advantage of the nonlinear representation ability of the more mature neural network and have robust and generalizability in cross-condition, especially for the signal process problem of nonlinear non-stationary vibration. From many kinds of literature, ML-based investigations [21]–[26] usually can get high performance. [15] divided the process of RUL estimation into four stages: data acquisition, health indicators (HIs) construction, health status (HS) assessment, and RUL estimation. ML-based RUL estimation methods are divided into two categories. The first category, called two-stage methodology, is to predict the bearing operating status and RUL by constructing HI as an intermediate bridge [27]. The second category, called one-stage methodology, is to use a black-box model in an end-to-end manner with source data as input and with predicted RUL as output [23]. Although many studies have obtained good results, because vibration data are sometimes accompanied by not only one type of failure, but also several types of failures at the same time, and sometimes vibration data are accompanied by some cyclic noises, which seriously interfere with the characterization of the degradation state. Therefore, it is difficult to directly capture the law of degradation changes over time from raw and unprocessed data by using a simple black-box model with regression prediction capabilities. For some two-stage approaches, the traditional extraction of frequency domain and time domain depends on some pre-set thresholds and does not have an excellent performance in feature representation, which is not applicable enough for vibration frequency belongs to nonlinear non-stationary characteristics. The importance of HI construction has also been confirmed in many published documents [14], [26], [28], [29]. To our best knowledge, the current study in bearing RUL estimation mainly faces the following difficulties:

(1)Some pre-set thresholds need to be determined manually. Most RUL investigations require bearing failure threshold as a prerequisite when extracting physical features, which researchers set manually in advance and limits their application in real scenarios.

(2)Objects investigated are always under a single (or

Manuscript received July 12, 2021; revised December 6, 2021. This work was supported by the National Key RD Program of China under grant No.2019YFB2102500 and the Science Foundation of Beijing-Shanghai High Speed Railway Co.,Ltd under grant No.2019166.(Corresponding author: Dongdong Zhao.)

D. Zhao is a PhD candidate of Beijing Jiaotong University, No.3 Shangyuancun Haidian District, Beijing, China, 100044 (e-mail: ddzhao@bjtu.edu.cn).

F. Liu is a Professor of Computer and Information Technology Department, Beijing Jiaotong University, No.3 Shangyuancun Haidian District, Beijing, China, 100044 (e-mail: fliu@bjtu.edu.cn).

X. Wang is a PhD candidate of Beijing Jiaotong University, No.3 Shangyuancun Haidian District, Beijing, China, 100044 (e-mail: wangxiaodong@bjtu.edu.cn).

same) operating condition. Many studies on bearing RUL estimation are based on a single operating condition, which always are not available when applied in the practical industry due to these investigations can't endure the shift between varying operating conditions, that is, to some extent, their performance can be easily shrunk.

(3)Not all samples in the entire life span of each test unit are observed and estimated. Under normal circumstances, researchers in industry and academia start to trigger the behavior of RUL estimation by default through first predicting time (FPT) detection, because the degradation characteristics of the initial state before the FPT point are so small or not obvious that the degradation characteristics cannot be captured. Therefore, the samples before the FPT point are usually discarded and no RUL estimation is made. Unfortunately, some research documents do not perform RUL estimation for each sampling moment of bearing during the RUL estimation but look for several points, and the final accuracy is based on the average of these points. In addition, Some studies focus on the middle stage and ignore the late stage for the last 10% vibration signal frequency is too violent, specially tested in the public dataset that the laboratory test-bed stops when reaching a certain vibration frequency before a laboratory bed collapses, which means the data of late-stage don't involve in the training model, so RUL estimation also can't be accomplished in practical projects, which means that it will be difficult to put these researches into practical applications.

The three problems mentioned above all seriously impede the transformation of research results in practical engineering projects. To solve these issues mentioned above, generalization and high practicality are regarded as key considerations in this investigation. In this paper, a novel generalized framework based on neural networks is investigated for RUL estimation. The main contributions of this paper are listed as follows:

(1)Dense low-dimensional HIs are obtained by combining with convolutional neural networks (CNN) and deep encoder (DE). CNN is utilized to extract latent features and avoid using preset thresholds. Encoders are proposed to get dense low-dimensional HIs instead of one-dimensional HI to avoid lossing effective information, that is, one sample is corresponding to specific HIs with a few elements. The proposed method in this paper strongly combining with performance advantages of CNN and DE to automatically construct a HI with a monotonous trend.

(2)Time series information are learned from dense HIs to predict RUL value. On the basis that GRU can alleviate the problem of gradient disappearance and has relatively few parameters, which reduces the risk of model overfitting. To explore the applicability and versatility of the proposed framework, the test data not only includes data with the same distribution as the training data but also data with a different distribution from the training data of varying operating conditions.

(3)The RUL corresponding to each sample from health to failure in each test bearing unit is accurately predicted in small error loss as possible through the proposed method called CDEGRU. Moreover, for further verification, the proposed method is performed on two platforms experimental verification.

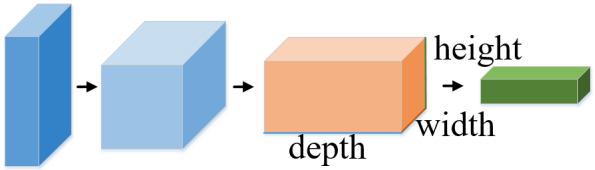


Fig. 1: The architecture of CNN.

The remainder of this paper is organized as follows. The related works and network structure of the proposed method are presented in Section II and Section III. In Section IV, dataset, experimental setup, experimental results, and analysis will be given. Then this paper is closed with conclusions in Section V.

II. Related Works

A. Convolutional Neural Networks

The traditional CNN consists one or more convolutional layer(s) and fully connected layer(s). Compared with other deep learning structures, CNN can obtain attractive results in image and speech recognition, making it an exciting deep learning structure. CNN usually contain the following contents:

1) Convolutional layer: Each convolutional layer in CNN has a number of convolutional units, and the parameters of each convolutional unit are obtained through optimization of the backpropagation algorithm. The purpose of convolution operation is to extract different scale latent high-level representations from raw data. The first or shallower convolutional layer may only extract some low-level features such as edges, lines, and corners. As the layer become more deeper, networks will dig more complex features from the low-level and abstract representations, which also means that, to some extent, as the network deepens, objects that seem to be more inscrutable can be fitted.

2) Activation function: The activation function operates on each node of each layer of nerves, which is an indispensable part of the convolution process.

3) Pooling layer: The pooling layer uses a maximum or average sampling mechanism, where the former emphasizes the salient elements of the sample to reduce the image size, and the latter smoothes and blurs the data to reduce the image resolution. To some extent, they also inhibit over-fitting.

4) Fully connected layer: The fully connected layer aggregates the acquired features, and then ranks the importance of the features in order to provide high-quality assistance for subsequent tasks.

CNN is used by many researchers in the field of studies on bearing to extract latent features [18], [24]–[26], [30], which proves the effectiveness of feature extraction on vibration data.

B. Deep Auto-encoder

The auto-encoder, consisting of encoder and decoder, is a type of artificial neural network used in semi-supervised learning and unsupervised learning. In the traditional linear

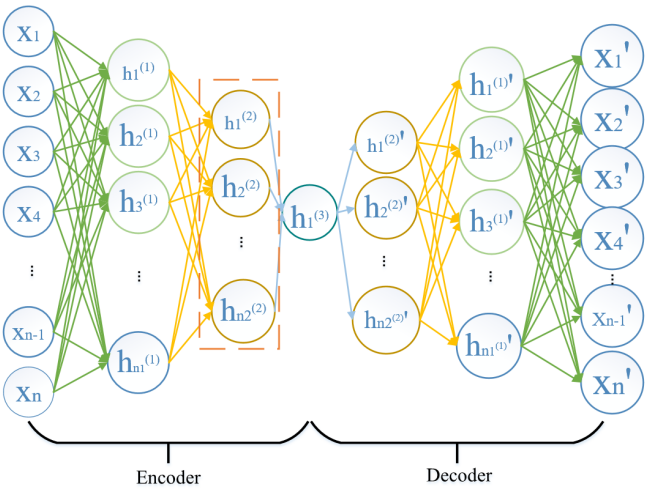


Fig. 2: The auto-encoder network.

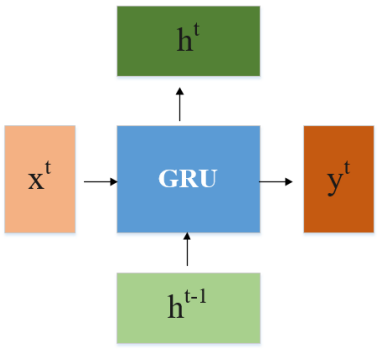


Fig. 3: The structure of GRUs.

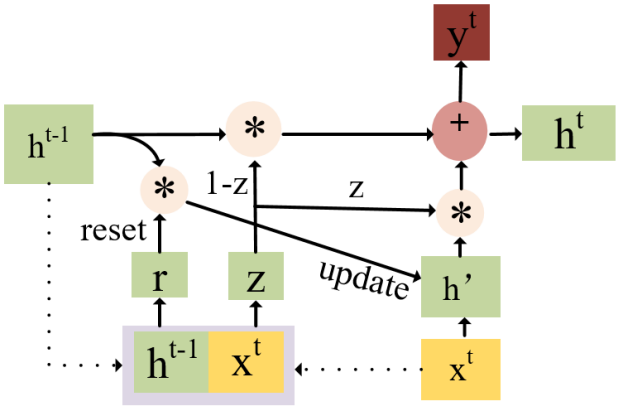


Fig. 4: The inner structure of GRUs.

case, researchers prefer to use principal component analysis (PCA) to reduce dimension, but for unstable nonlinear vibration data, PCA is powerless, ISOMAP [15] are used to reduce one dimensional HI under the eight extracted features. The deep auto-encoder compresses the input data into a latent spatial representation, which is represented by $h = f(x)$. This part of the decoder reconstructs the input from the latent representation, which are represented by the function $s = g(h)$, that is, $s = g(f(x))$. Where x is the input data, f and g are respectively the map function of encoder and decoder. Therefore, the function of the encoder is to learn the input information by taking the input information as the learning target. It is usually used in dimension reduction, noise reduction, and outlier detection research [31], [32]. The network architecture of the deep auto-encoder is shown in the Fig.2. The data compression is mainly completed in two stages. The first stage is the encoding stage. The input vector $x = (x_1, x_2, x_3, ..., x_n)$ is compressed into $h_1 = (h_1^{(1)}, h_2^{(1)}, h_3^{(1)}, ..., h_{n_1}^{(1)})$ by the 1-th hidden layer of encoder. $h_i^{(j)}$ represents the i-th node value of the j-th hidden layer. n_1 of $h_{n_1}^{(j)}$ is equal to the number of neural nodes in the j-th hidden layer. Then n_1 is compressed as $h_2 = (h_1^{(2)}, h_2^{(2)}, h_3^{(2)}, ..., h_{n_2}^{(2)})$, then h_2 is compressed as $h3 = (h_1^{(3)})$ by the 2-th hidden layer of encoder, then decoded as $h_1' = (h_1^{(1)}, h_2^{(1)}, ..., h_{n_1}^{(1)})$ by the first hidden layer of decoder, and then $h_2' = (h_1^{(2)}, h_2^{(2)}, ..., h_{n_2}^{(2)})$ decoding by the second hidden layer of decoder, and finally decoded as $x = (x_1', x_2', ..., x_n',)$, which is the obtained output of auto-encoder and the forward propagation process, marked as $h_{w,b}(x)$, and then the parameters will be updated by the mean of gradient learning in backward propagation $h_{w,b}(x) = \frac{1}{m}\sum_{i=1}^{m}(\hat{x} - x)^2$. By using the gradient optimization method, for example, stochastic gradient descent [33], Adam [34], Adagrad [35], the network weights w, b are optimized to reduce the value of the loss function of the forward propagation.

C. Gated Recurrent Units

Gated Recurrent Units (GRUs) is a type of recurrent neural network (RNN). Compared with LSTM, GRUs can achieve considerable results, which is easier to train and can greatly improve the efficiency of training process. The structure of GRUs shown in Fig.3. is the same as that of ordinary RNN. x_t is a current input, and the hidden state is passed from the previous node h_{t-1}, which contains information about the previous node. Combining x_t with h_{t-1}, GRUs will get the output of the currently hidden node and the hidden state passed to the next node.

First, we obtain the two gating states through the last transmitted state h_{t-1} and the current node input x_t. As shown in Fig.4., r is the gating to control reset and z is the gating to control to update.

$$\begin{aligned} r &= \sigma(W^r[h^{t-1}, x^t]) \\ z &= \sigma(W^z[h^{t-1}, x^t]) \end{aligned} \tag{1}$$

Where σ is a activation function, through which the data can be transformed into a value in the range of 0 and 1, thus acting as a gating signal. After getting the gating signal, first use reset gating r to get the data $h_{t-1}' = h_{t-1}r$, then splice h_{t-1}' with the input x_t, finally a activation function $tanh$ is used to shrink the data obtained by splicing to the range of -1 and 1.

$$h' = tanh(W[h^{t-1\prime}, x^t]) \tag{2}$$

Where h' mainly includes remembered last moment h_{t-1} and the current input x_t.

'*' in Fig.4. represents Hadamard Product, which means that the corresponding elements in the operation matrix are multiplied, so it is required that the two multiplication matrices are of the same type. '+' represents the matrix addition operation.

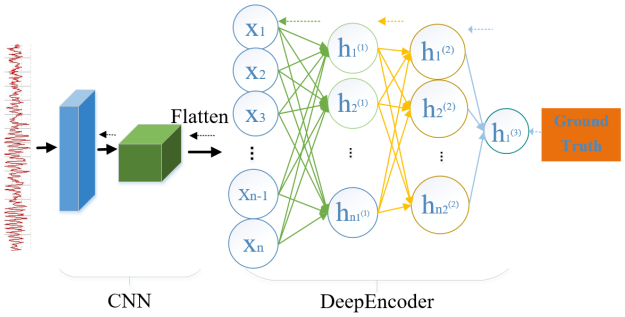


Fig. 5: The flowchart of stage 1 in CNN-DeepEncoder.

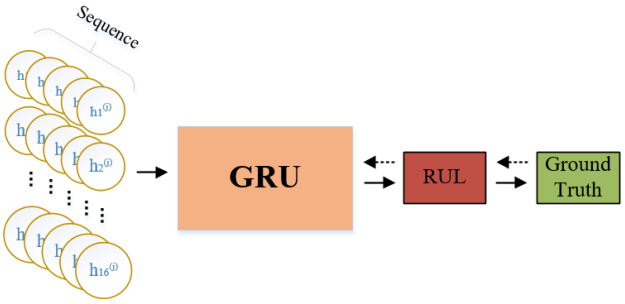


Fig. 6: The flowchart of GRUs in stage 2.

Updating memory is the critical step. Unlike LSTM, GRUs uses the same gating z for forgetting and selective memory.

$$h^t = z * h^{t-1} + (1-z) * h' \qquad y^t = \sigma(W h^t) \tag{3}$$

Where former part indicates selective "forgetting" of the original hidden state and forgetting some unimportant information, and the latter part means memorizing the information containing the current node. y_t is the output of the GRUs. It is worth mentioning that the value of z ranges from 0 to 1, and the larger the value, the larger the weighting factor of the hidden state at the previous moment.

Overall, GRUs have one less internal parameter for gating. Therefore, in practical applications, considering the computing power and time cost of hardware, most scholars will choose the more practical GRUs to engage in related research [36]–[38].

This section has presented the foundation needed for the proposed network. The next section will introduce the detailed architecture and implementation of the proposed method in this paper.

III. Proposed Method

This section mainly introduces the network structure of the proposed algorithm, which is divided into two parts: stage 1 and stage 2. The purpose of stage 1 is the feature extraction and dimension reduction for HI construction, and the purpose of stage 2 is to learn time-series information from the result of stage 1 and to output an estimated RUL closer to the actual label as possible. The Fig.7. shows the learning flowchart of the whole process. In this paper, the proposed method is called CNNDeepEncoder+GRUs (abbreviated as CDEGRU). The CNNDeppEncoder stage 1 is offline training.

1) Stage 1: Stage 1, also called CNN-DeepEncoder, is effective feature extraction and dimension reduction method. Combined with nonlinear representation ability of CNN [21], [24]–[26], [30] and dimension reduction ability of encoder [31], [32], so as to better convert the nonlinear vibration data to dense low-dimensional HIs. Fig.5. is the architecture diagram of the CNN-DeepEncoder stage of the proposed algorithm. The solid line is the forward propagation, and the dashed line is the backward propagation.

The stage 1 HI construction contains convolutional layers C1 and C2, pooling layers P1 and P2, and four

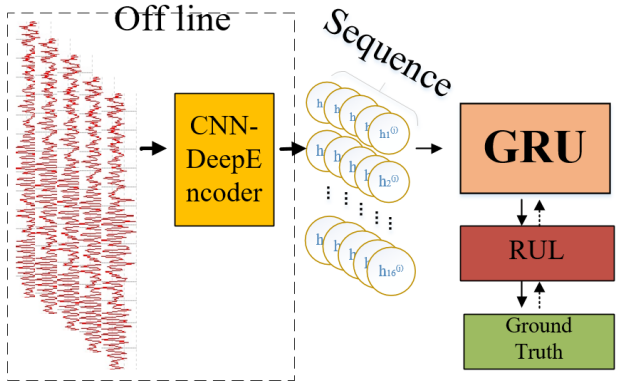


Fig. 7: The whole flowchart of method proposed in this paper.

fully-connected layers F1, F2, F3, and F4. The raw data are input and passed into the first convolutional layer C1 and max-pooling layer P1, and in turn output of P1 is passed into the C2. After the output of P2 is flatten, then in turn the features extracted are fed into the fully connected layer F1, F2, F3, F4, which constructs into HI containing a single element.

2) Stage 2: Based on stage 1, GRUs are used in stage 2 for time series learning. GRUs contain a pooling layer and two GRUs layers, generates a predicted RUL, then calculates the error loss with the actual RUL, and then performs backward propagation to update parameters. The input of the GRUs is the specific series of extracted feature data of the penultimate layer of stage 1, which obtain an RUL predicted value, and then through loss calculation and post-propagation gradient learning to narrow the value between the predicted RUL and the actual RUL. The Fig.6. shows the training flowchart of the GRUs in stage 2. The solid line indicates forward propagation, and the dashed line indicates backward propagation. Overall, the value of neural nodes in the CNN-DeepEncoder stage 1 are connected into a series of data, which fed into the GRU model that outputs a predicted RUL, then loss between the predicted RUL and the actual RUL are calculated. Finally, the loss is minimized after many iterations to achieve parameters optimization.

IV. Experimental Study

This chapter includes the experimental datasets, implementation details, compared approaches, and the experimental result and analysis. Experimental results are made

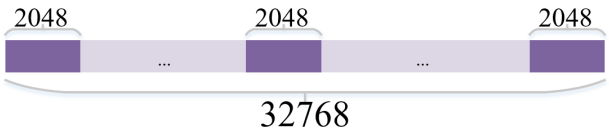


Fig. 8: Data augmentation in this study.

up of prognostic performance, feature visualizations, and the effect of the time window and batch size.

A. Dataset

The proposed method is verified on two datasets: FEMTO [39] and XJTU_SY dataset [22].

1) FEMTO [39]: , FEMTO dataset was collected on the PRONOSTIA that consists of three main parts: a rotating part, a degradation generation part, and a measurement part. This dataset includes 17 run-to-failure bearings under varying three conditions as shown in Table I. For each bearing, 2560 data points are collected on 0.1 s of every 10 s with a sampling frequency of 25.6 kHz. For more degradation information, this study also uses two-directional data. The training datasets are the first two bearing in each condition. The test datasets are the remaining bearing, where Ai-j represents the j-th bearing of i-th condition on the FEMTO.

TABLE I: Information of the FEMTO.

Condition	Load(N)	Speed(rpm)	Name
1	4000	1800	A1-1, A1-2, A1-3, A1-4, A1-5, A1-6, A1-7
2	4200	1650	A2-1, A2-2, A2-3, A2-4, A2-5, A2-6, A2-7
3	5000	1500	A3-1, A3-2, A3-3

2) XJTU-SY dataset: The XJTU-SY dataset was collected by the Xi'an Jiaotong University and the Changxing Sumyoung Technology Company [22]. 32768 data points are collected on 1.28 s of every minute with a sampling rate of 25.6 kHz. The process of tests of bearing is stopped when the amplitude of the vibration signal is higher than 20 g for protecting the testbed. There are two PCB 352C33 accelerometers are placed on the housing of the tested bearings, which are respectively on the vertical and horizontal axis. For more degradation information, this study also uses two-directional data. The information of the XJTU-SY dataset is shown in Table II. In this study, the training datasets are B1-1/3/4/5, B2-1/5. The test datasets are B1-2/2/3/4, where Bi-j represents the j-th bearing of i-th condition on the XJTU-SY dataset.

TABLE II: Information of XJTU-SY Dataset

Condition	Radial force(N)	Rotating speed(rpm)	Name
1	12000	2100	B1-1, B1-2, B1-3, B1-4, B1-5
2	11000	2250	B2-1, B2-2, B2-3, B2-4, B2-5

B. Data Preprocessing and Performance Metrics

In this study, the data augmentation mechanism shown in Fig.8. is applied on XJTU-SY. A recorded sample with 32768 data points was evenly cut into 16 samples with 2048 data points. For the 16 samples of a recorded sample with 32768 data points, every one sample with 2048 data points participate in training as an independent sample, and their ground truth is the same as each other. When verified, the recorded sample's predicted RUL is the average of 16 sub-samples.

The experimental environment is based on the configuration of the GTX1050Ti graphics card and the programming language is python. Mean Square Error (MSE) is used as a loss function. Adam optimization algorithm in the back-propagation learning process is used for the update of model parameters, and other relevant parameters are as shown in the Table III.

TABLE III: The parameters used in CDEGRU.

Parameters	Value	Parameters	Value
Epochs in stage 1	300	Learning rate	1e-3
Epochs in stage 2	500	Batch size	32
Dropout	0.15	N_{tw}	50

Performance metrics are the most important consideration in experimental research, and their choice directly determines the scientificity and fairness of experimental research. In this experimental study, three performance evaluation criteria were used to measure the performance of the experiment: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). Suppose the predicted value is $\hat{y} = \{\hat{y}_1, \hat{y}_2, \hat{y}_3, ..., \hat{y}_n\}$ and actual value is $y = \{y_1, y_2, y_3, ..., y_n\}$. MAE is the mean absolute error, which can better reflect the actual situation of the predicted error. The definition of MAE is shown in Eq. (4).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y_i} - y_i| \tag{4}$$

RMSE, which measures the deviation between the observed value and the true value. Commonly used as a standard for measuring the prediction results of machine learning models. The definition of RMSE is shown in Eq. (5).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2} \tag{5}$$

MAPE is used to evaluate the errors of different models in the same data set. The definition of MAPE is shown in Eq. (6).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y_i} - y_i}{y_i}\right| \tag{6}$$

C. Compared Approaches

To explore the advantages and disadvantages of feature extraction and dimension reduction between CNN, DNN, and encoder. This paper proposes to use the following five methods as baselines to compare with the proposed method. CNN contains five convolutional layers and three fully connected layers. DNN contains two convolutional layers and eight fully connected layers. To maintain a fair comparison as much as possible, the input and output

have the same size in stage 1 and stage 2. The algorithms compared in this paper are as follows:

(1)CNN: Only CNN in stage 1 and no stage 2. The output of the last layer of CNN in stage 1 is the estimated RUL.

(2)DNN: Only DNN in stage 1 and no stage 2. The output of the last layer of DNN in stage 1 is estimated RUL.

(3)CDE: The CDE method is implemented as one of baseline. Specifically, stage 1 have CNN and the encoder, but no stage 2. The output of the last layer of the encoder in stage 1 is the estimated RUL.

(4)CGRU: The CNN_GRU method, including CNN of stage 1 and GRUs of stage 2, is implemented as one of baseline. Specifically, the input of GRUs in stage 2 is from the output of the penultimate layer of CNN in stage 1.

(5)DGRU: The DNN_GRU method, including DNN in stage 1 and GRUs in stage 2, is implemented as one benchmark. Specifically, GRUs' input comes from DNN's output in stage 1.

D. Experimental Results

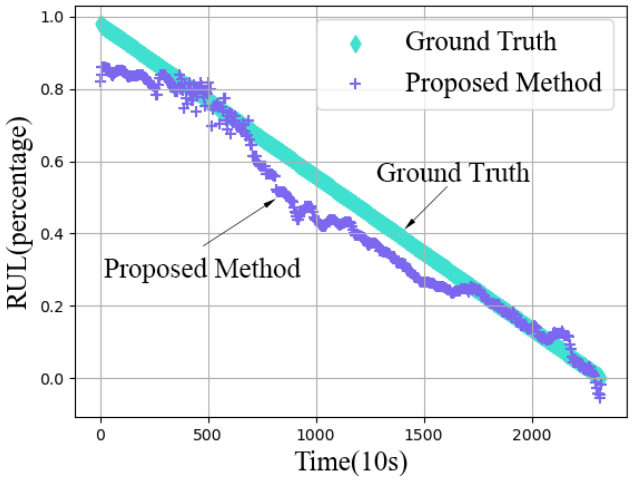
This part gives experimental results and performance analysis, mainly including prognostic performance, feature visualizations, and effects of varying time windows.

1) Prognostic performance: The main content of this part is the verification results of the proposed experiment CDEGRU and baseline experiment. The verification results are shown in Table IV and Table V. The last column 'ave' represents the average of all test sub-datasets. Because the value range of RUL is between 0 and 1, the values of MAE and RMSE are amplified 100 times larger than the actual result for a clear view. Note that when verified on XJTU-SY, the number of training iterations in stage 1 is 200, the number of training iterations in stage 2 are 300, the batch size is 16, and other parameters are same as on FEMTO'.

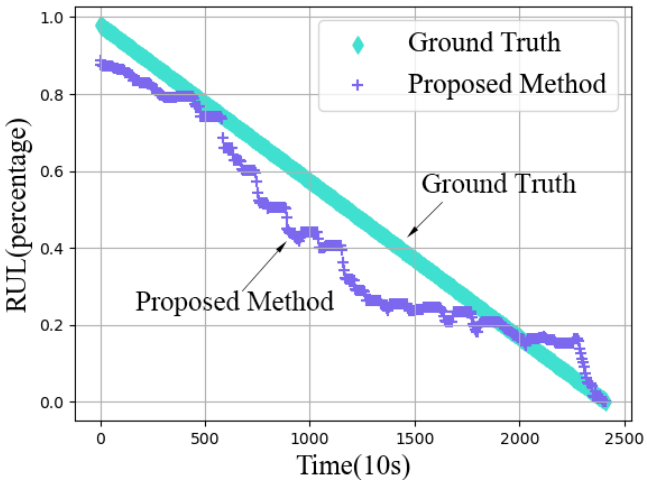
From Table IV and Table V, CDE is better than CNN and DNN, CDEGRU is better than CDE, CGRU, and DGRU. In the test sub-datasets of each column, almost the smallest MAE and RMSE come from the CDEGRU. It shows that CDE has considerable advantages in nonlinear feature extraction and dimension reduction. And GRUs play an indispensable role in time series information learning. According to MAPE in Table IV and Table V, it turns out the proposed method is better than other baselines, especially for Table V the performance on the XJTU-SY are more significant.

Fig.9. gives out the result of RUL estimation on two bearings A1-4 and A1-5 of FEMTO.

Note that in Fig.9 the abscissa represents time, the unit is 10 s, and the ordinate is RUL percentage. In Fig.9., the estimated RUL of A1-4 and A1-5 are basically kept near and below the ground truth, and A1-4 and A1-5 are all almost close to actual RUL before failure. A1-4 shown in Fig.9(a) appears to have two stages. The first stage is probably between 0 and 500 of the life span, followed by the second stage. Specifically, in the first stage, the estimated value and the true value are almost kept at a similar slope, and for the last stage, the estimated value



(a) A1-4



(b) A1-5

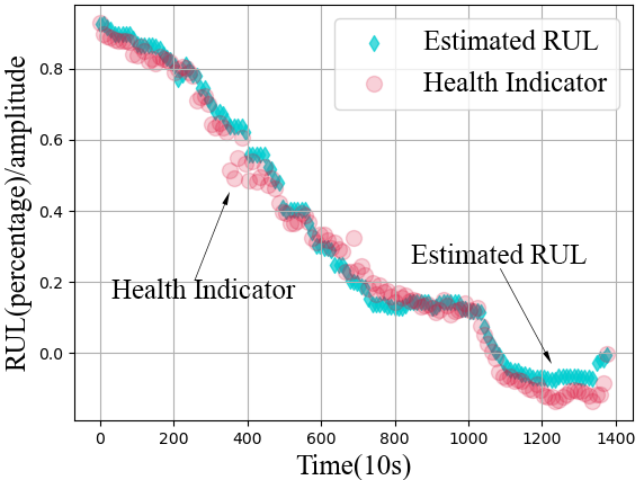
Fig. 9: The RUL estimation on FEMTO:(a)A1-4(b)A1-5.

and the true value basically coincide. Different from A1-4, A1-5 shown in Fig.9(b) appears to have three stages. In the initial stage, which is the first quintile of the near life span, the estimated values remain between 0.85 and 0.78 and slowly decrease. In the middle of the second and third quintiles, the estimated value began to decrease. Although in this range the estimated value is lower than the true value, it can provide on-site engineers with early warnings so as not to be too late to overhaul the bearings. After that is the last stage, the estimated value basically fluctuates around the true value. Compared with the initial and intermediate stages, the accuracy of the last stage is higher.

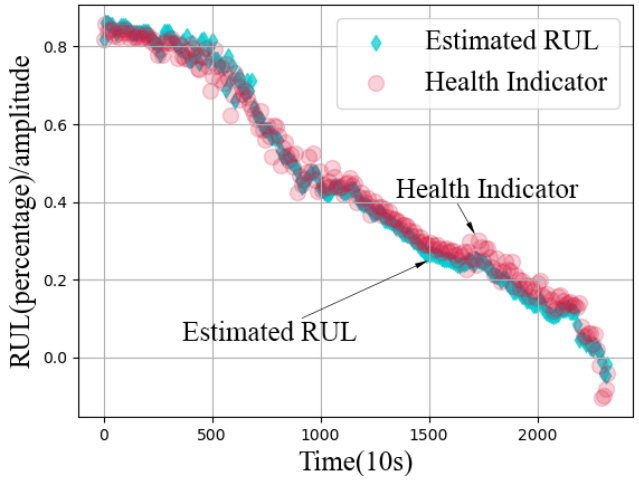
2) Feature visualizations: The data of feature visualizations in this section come from the one-dimensional HI of the last layer in stage 1 and the HI of GRUs in stage 2. Fig.10. gives estimated RUL and HI of A1-3 and A1-4. From Fig.10, we can see that whether it is A1-3 or A1-4, the HI fluctuates larger than the estimated RUL, and RUL estimation inherits HI construction's degeneration trend. It is more obvious in Fig.10(a) that the estimated RUL are closer to the true degradation situation than the HI, such as between 200 and 300 and between 1000 and 1400 of life span.

TABLE IV: Performance comparisons of different methods on the FEMTO.

Methods	Metrics	Sub-datasets											
		A1-3	A1-4	A1-5	A1-6	A1-7	A2-3	A2-4	A2-5	A2-6	A2-7	A3-3	Ave
CNN	MAE	10.5	20.5	13.2	13.3	16.3	37.8	40.3	11.7	15.0	41.0	22.1	22.0
	RMSE	13.4	25.3	16.3	16.2	21.6	43.2	47.5	14.3	19.4	44.3	26.0	26.1
	MAPE	27.2	32.9	27.2	25.4	32.5	10.6	9.6	26.0	20.1	16.8	20.7	22.6
DNN	MAE	5.5	15.7	13.7	17.3	14.3	30.5	37.0	19.3	11.7	42.7	21.6	20.8
	RMSE	6.4	19.4	16.7	19.9	19.9	35.9	42.1	24.8	15.0	45.6	24.1	24.5
	MAPE	26.3	32.7	25.3	24.0	32.2	10.1	5.5	13.5	19.2	17.8	18.7	20.5
CDE	MAE	6.9	14.3	11.5	15.6	13.8	31.3	35.2	15.6	10.2	38.8	19.6	19.3
	RMSE	8.8	17.6	14.5	18.5	18.5	37.3	42.1	19.9	14.0	42.4	23.6	23.4
	MAPE	25.7	32.1	25.1	22.4	30.4	11.0	10.6	18.7	19.7	16.0	19.8	21.0
CGRU	MAE	5.2	18.4	15.0	12.5	16.3	34.3	37.1	12.2	13.7	42.5	17.4	20.4
	RMSE	6.5	22.1	17.6	13.8	20.8	39.4	43.4	14.2	16.8	45.6	20.5	23.7
	MAPE	25.8	31.7	26.2	24.5	32.3	8.5	3.5	24.4	16.6	12.4	22.5	20.8
DGRU	MAE	5.9	15.8	14.5	14.2	13.6	26.5	34.8	20.2	13.3	36.8	23.4	19.9
	RMSE	7.8	19.6	17.7	17.1	17.9	33.2	39.6	26.3	18.7	40.6	26.7	24.1
	MAPE	24.3	31.5	24.4	22.2	29.3	9.8	9.1	18.9	18.5	14.2	17.5	20.0
CDEGRU	MAE	4.6	12.8	8.4	13.2	14.7	29.6	32.0	8.6	7.7	38.9	19.5	17.3
	RMSE	5.9	14.7	9.6	18.9	19.7	34.0	38.6	10.8	9.9	42.0	21.9	20.6
	MAPE	25.6	31.0	25.0	22.2	29.2	10.2	7.9	12.0	17.2	11.8	18.4	19.1



(a) A1-3



(b) A1-4

Fig. 10: The indicator of stage 1 and predicted RUL in stage 2 for two bearings on FEMTO: (a) A1-3 and (b) A1-4.

TABLE V: Performance comparisons of different methods on the XJTU dataset.

Method	Metrics	Sub-datasets				
		B1-2	B2-2	B2-3	B2-4	Ave
CNN	MAE	22.9	18.3	20.2	27.8	22.3
	RMSE	27.0	20.9	25.2	29.7	25.7
	MAPE	26.2	29.1	9.2	1.0	16.4
DNN	MAE	19.1	15.2	20.1	33.2	21.9
	RMSE	25.5	18.6	23.7	40.4	27.0
	MAPE	21.6	28.6	7.6	0.8	14.7
CDE	MAE	19.4	15.0	18.9	25.1	19.6
	RMSE	25.9	17.6	22.9	31.8	24.5
	MAPE	0.6	31.2	0.6	0.4	8.2
CGRU	MAE	22.2	16.7	21.1	21.7	20.4
	RMSE	26.9	19.6	24.5	26.6	24.4
	MAPE	15.9	24.9	5.5	0.8	11.8
DGRU	MAE	19.0	15.3	22.7	33.6	22.6
	RMSE	24.9	18.5	26.0	41.1	27.6
	MAPE	17.7	23.9	6.7	0.8	12.3
CDEGRU	MAE	17.4	14.0	19.5	14.6	16.4
	RMSE	22.1	17.2	23.2	17.7	20.1
	MAPE	2.3	28.2	0.5	0.9	8.0

3) Time window: In the experimental implementation on the two public datasets, the pre-set time window N_{tw} of GRUs in stage 1 and stage 2 respectively are 30, 50, and 100. In addition, the value of the time window is pre-set, it only represents the max sequence length. Because before training, the sample sequence length is a random number from half of N_{tw} to N_{tw}. When the length doesn't exactly equal to N_{tw}, the remaining part(s) that N_{tw} minus the random number will be replaced by zero. Therefore, the size of the time window is not a fixed length in the absolute sense.

TABLE VI: Parameters used in CDEGRU.

Time windows	MAE	RMSE
30	19.3	22.9
50	19.2	22.5
100	21.4	24.3

TABLE VII: Parameters used in CDEGRU.

Time windows	MAE	RMSE
30	24.4	29.4
50	16.4	20.1
100	20.6	24.8

The results in Table VI are the average on sub-datasets of FEMTO. The results in Table VII are the average on sub-datasets of the XJTU-SY dataset. Table VI and Table VII show the compared result about different time windows of GRUs in stage 1 and stage 2. From these two tables, 50 is a more suitable parameter. Too short or too long will get worse MAE and RMSE.

V. Conclusion

The goal of this investigation is how to improve generalization ability and high practicality. In this paper, CNN, encoder, and GRU technology are investigated to estimate RUL for the rotating machine. CNN is used to extract latent features. An encoder is used to reduce the dimension of latent features to dense low dimensional space. For GRU is used to learn time-series information. The proposed method obtains desirable results and is promising for industrial applications. Though the proposed method has an attractive performance, some shift between varying platform or complexity operating conditions, that is, data from some different distributions, ought to be paid more attention. Thus, we are devoting ourselves to RUL estimation via domain adaptation.

References

[1] J. Singh, M. Azamfar, F. Li, and J. Lee, "A systematic review of machine learning algorithms for prognostics and health management of rolling element bearings: fundamentals, concepts and applications," *Measurement Science and Technology*, vol. 32, no. 1, p. 012001, 2020.

[2] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to rul prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.

[3] D. Zhao, F. Liu, and H. Meng, "Bearing fault diagnosis based on the switchable normalization ssgan with 1-d representation of vibration signals as input," *Sensors*, vol. 19, no. 9, p. 2000, 2019.

[4] X. Wang, F. Liu, and D. Zhao, "Cross-machine fault diagnosis with semi-supervised discriminative adversarial domain adaptation," *Sensors*, vol. 20, no. 13, p. 3753, 2020.

[5] B. Zhang, C. Sconyers, C. Byington, R. Patrick, M. E. Orchard, and G. Vachtsevanos, "A probabilistic fault detection approach: Application to bearing fault detection," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 2011–2018, 2010.

[6] J. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1803–1836, 2011.

[7] C. J. Lu and W. O. Meeker, "Using degradation measures to estimate a time-to-failure distribution," *Technometrics*, vol. 35, no. 2, pp. 161–174, 1993.

[8] W. Q. Meeker, L. A. Escobar, and C. J. Lu, "Accelerated degradation tests: modeling and analysis," *Technometrics*, vol. 40, no. 2, pp. 89–99, 1998.

[9] K. A. Doksum and A. Hbyland, "Models for variable-stress accelerated life testing experiments based on wener processes and the inverse gaussian distribution," *Technometrics*, vol. 34, no. 1, pp. 74–82, 1992.

[10] J. M. van Noortwijk, "A survey of the application of gamma processes in maintenance," *Reliability Engineering & System Safety*, vol. 94, no. 1, pp. 2–21, 2009.

[11] X. Wang and D. Xu, "An inverse gaussian process model for degradation data," *Technometrics*, vol. 52, no. 2, pp. 188–197, 2010.

[12] J. P. Kharoufeh, "Explicit results for wear processes in a markovian environment," *Operations Research Letters*, vol. 31, no. 3, pp. 237–244, 2003.

[13] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–202, 1972.

[14] C. Cheng, G. Ma, Y. Zhang, M. Sun, F. Teng, H. Ding, and Y. Yuan, "Online bearing remaining useful life prediction based on a novel degradation indicator and convolutional neural networks," *arXiv preprint arXiv:1812.03315*, 2018.

[15] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Remaining useful life estimation based on nonlinear feature reduction and support vector regression," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1751–1760, 2013.

[16] Z. Tian, "An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring," *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 227–237, 2012.

[17] A. Elsheikh, S. Yacout, and M.-S. Ouali, "Bidirectional handshaking lstm for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, 2019.

[18] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.

[19] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliability Engineering & System Safety*, vol. 183, pp. 240–251, 2019.

[20] A. Zhang, H. Wang, S. Li, Y. Cui, Z. Liu, G. Yang, and J. Hu, "Transfer learning with deep recurrent neural networks for remaining useful life estimation," *Applied Sciences*, vol. 8, no. 12, p. 2416, 2018.

[21] W. Mao, J. He, J. Tang, and Y. Li, "Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network," *Advances in Mechanical Engineering*, vol. 10, no. 12, p. 1687814018817184, 2018.

[22] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401–412, 2018.

[23] R. Khelif, B. Chebel-Morello, S. Malinowski, E. Laajili, F. Fnaiech, and N. Zerhouni, "Direct remaining useful life estimation based on support vector regression," *IEEE Transactions on industrial electronics*, vol. 64, no. 3, pp. 2276–2285, 2016.

[24] Z. Chen, X. Tu, Y. Hu, and F. Li, "Real-time bearing remaining useful life estimation based on the frozen convolutional and activated memory neural network," *IEEE Access*, vol. 7, pp. 96 583–96 593, 2019.

[25] Y. Yoo and J.-G. Baek, "A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network," *Applied Sciences*, vol. 8, no. 7, p. 1102, 2018.

[26] L. Guo, Y. Lei, N. Li, T. Yan, and N. Li, "Machinery health indicator construction based on convolutional neural networks considering trend burr," *Neurocomputing*, vol. 292, pp. 142–150, 2018.

[27] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, "Data alignments in machinery remaining useful life prediction using deep adversarial neural networks," *Knowledge-Based Systems*, vol. 197, p. 105843, 2020.

[28] L. Guo, H. Gao, H. Huang, X. He, and S. Li, "Multifeatures fusion and nonlinear dimension reduction for intelligent bearing condition monitoring," *Shock and Vibration*, vol. 2016, 2016.

[29] D. Zurita, J. A. Carino, M. Delgado, and J. A. Ortega, "Distributed neuro-fuzzy feature forecasting approach for condition monitoring," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 2014, pp. 1–8.

[30] G. S. Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *International conference on database systems for advanced applications*. Springer, 2016, pp. 214–228.

[31] Y. Hu, T. Palmé, and O. Fink, "Deep health indicator extraction: A method based on auto-encoders and extreme learning machines," in *PHM 2016, Denver, USA, 3-6 October 2016*. PMH Society, 2016, pp. 446–452.

[32] L. Ren, Y. Sun, J. Cui, and L. Zhang, "Bearing remaining useful life prediction based on deep autoencoder and deep neural networks," *Journal of Manufacturing Systems*, vol. 48, pp. 71–77, 2018.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[34] K. Da, "A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.

[36] Y. Duan, H. Li, M. He, and D. Zhao, "A bigru autoencoder remaining useful life prediction scheme with attention mechanism and skip connection," *IEEE Sensors Journal*, 2021.
[37] L. Ren, J. Cui, Y. Sun, and X. Cheng, "Multi-bearing remaining useful life collaborative prediction: A deep learning approach," *Journal of Manufacturing Systems*, vol. 43, pp. 248–256, 2017.
[38] D. She and M. Jia, "A bigru method for remaining useful life prediction of machinery," *Measurement*, vol. 167, p. 108277, 2021.
[39] E. Sutrisno, H. Oh, A. S. S. Vasan, and M. Pecht, "Estimation of remaining useful life of ball bearings using data driven methodologies," in *2012 ieee conference on prognostics and health management*. IEEE, 2012, pp. 1–7.