Using temporal convolution network for remaining useful lifetime prediction

Jungan Chen | Danjiang Chen | Gaoping Liu

Abstract
Because Convolutional Neural Network (CNN) can extract spatial feature, while Long Short-Term Memory (LSTM) can learn temporal features, many methods combining CNN with LSTM are proposed for remaining useful lifetime prediction. In practice, it is better to learn temporal features from long history sequence data because of the slow inherently long-term degradation process. However, LSTM is less efficient in processing long history sequence. To solve this problem, in this work, a temporal convolution network combining causal filters with dilated convolutions is used to expand the receptive field length of network. The network structure can be fixed through three key parameters, and the size of time window adopted for time sequence processing is the same as the receptive field length. These two characteristics allow the network to easily be applied for engineering purposes. The method is tested and evaluated using two well-known datasets, namely the “Turbofan Engine Degradation Simulation Dataset C-MAPSS” and “Milling Dataset.” The performance analysis shows that the proposed method outperforms more classical methods in terms of prediction accuracy.

KEYWORDS
CNN, deep learning, LSTM, remaining useful lifetime prediction

1 | INTRODUCTION

With the development of Industrial Internet of Things, various sensors are installed for monitoring the operational behavior and health of machines. Exploiting these sensors data information for intelligent prognostic and health management (PHM) can improve the system reliability and reduce maintenance costs. In general, PHM is composed of four technical processes including data acquisition, health indicator (HI) construction, health stage division, and remaining useful life (RUL) prediction. Among them, RUL prediction can provide the guide information for predictive maintenance, which is very useful to increase the availability, reliability, and safety of systems. Therefore, it has attracted more and more attention in recent years. In general, RUL prediction approaches can be categorized into three categories: model-based, data-driven based, and hybrid approaches.

In model-based methods, mathematical models are building to describe the deterioration process and can be generally divided into two families, physics model-based methods and statistical model-based methods. The physics
model-based techniques, including Paris law, Forman law, Norton law, and so on, can predict the RUL accurately with enough knowledge about the physics of damage. However, this knowledge about some complex systems and new equipment is difficult to be completely understood. In contrast, statistical model-based methods to estimate the RUL of machinery are established based on empirical knowledge. These models include stochastic process, Markov process, Wiener process, Gaussian mixture model, particle filter, Eyring model. In Wiener process model, there are three different models, that is, the linear model, the exponential-like model, and the nonlinear model. However, under the complex and noisy working environments, it is difficult to create a robust mathematical model. In summary, model-based approaches cannot be suitable to all systems, and developing a specific model can be very costly.

Comparing with the model-based model, the data driven approaches can predict the RUL without knowing the physical nature of the degradation mechanism. In traditional data-driven approaches, human experts are usually participated to process and analysis of data which is time consuming. In contrast, the deep learning (DL) methods can automatically learn the features for diagnostic without human experts and become popular.

In the past few years, many typical DL frameworks are presented, which include Stacked Auto Encoder (SAE), Restricted Boltzmann machine (RBM), Deep Belief Network (DBN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and their variants have been developed in the field of machine health monitoring. More details are given in section of related work. According to Reference 30, these methods can be further divided into two main groups. One is direct mapping, another is similarity-based HI curve matching. In direct mapping methods, a model trained with the data collected under certain situation may not generalized to other situations, DBNs integrated with ensemble learning is proposed and multiobjective evolutionary algorithm is used to choose the most appropriate DBN models and related optimal parameters. As CNN is popularly applied in various areas, CNN together with time, frequency, and time-frequency domain feature is proposed for RUL estimation. As sensor data are time series data and RUL estimation model should be capable capture the time sequence information, RNN approaches are one suitable solution. But Traditional RNN has long-term time dependency problems, so Long Short-Term Memory (LSTM) with gate control approach is used. In curve matching method, CNN-based HI construction method, LSTM-based or Gated Recurrent Unit (GRU) autoencoder scheme are proposed to construct HI value.

Among these methods, CNN can auto extract the spatial feature related with the dependencies of sensor values through time-window approach. RNN/LSTM can extract the temporal feature related with time dependencies. Therefore, many methods combining CNN with LSTM are proposed because of the complementary strengths of CNN and LSTM.

As we know, change of RUL is a slow activity, so long data history is required to be learnt in LSTM. However, LSTM has less efficiency in processing long history sequence. In Reference 34, by combining causal filters with dilated convolutions, temporal convolution network (TCN) exhibit substantially longer memory, and are thus more suitable for domains where a long history is required. Furthermore, the models with causal convolutions do not have recurrent connections and always train faster than RNNs, especially when applied to very long sequences. Through experiments, TCN with minimal tuning outperforms LSTM/GRU of the same model size in most cases and has a better capability for long-term memory than other recurrent architectures.

Through above reviews, to process long history sequence and learn both temporal and spatial dependencies, TCN is designed for remaining useful lifetime prediction in this work. To the best of our knowledge, it is the first time to use TCN for the RUL prediction.

The main contribution of this work includes:

1. As the change of RUL is a slow activity, TCN model is designed to efficiently process such long history sequence. The network structure can be fixed through three key parameters which are the layer length, the kernel size, and number of filters. Therefore, it is easily applied in engineering. While the design of network structure such as CNN, LSTM are more complicated and to be reconsidered in different application fields.
2. The default size of time window can be calculated through other parameters. In common sense, it is easily accepted that the size of time window is just as the receptive field length. Therefore, the default size of time window adopted for time sequence processing is the same with the receptive field length which is calculated with the layer length and the kernel size in TCN.
3. Through the evaluation with two public dataset, the proposed method outperforms most classic methods. Furthermore, it is concluded that three evaluation metrics are needed to be used together.
2 | RELATED WORKS

Recent years, lots of DL methods including DBN, CNN, RNN/LSTM, and so on, are applied for RUL prediction.

2.1 | DBN

In Reference 13, a RBM-based method is used to predict the root mean square (RMS) in an unsupervised way. Linear regression layer was added at the top of RBM to predict the bearing RUL value. In their later work, DBN-Feedforward Neural Network (FNN) is proposed to predict RUL value directly by replacing RBM with DBN. In Reference 14, an enhanced RBM with new regularization term modeling is used to extract features and an unsupervised Self Organizing Map (SOM) algorithm aggregates the subset of features into a one-dimensional health value, which is used to predict RUL through a similarity-based life prediction algorithm. By integrating a multi objective evolutionary algorithm with the traditional DBN, a multi objective deep belief networks ensemble (MODBNE) method is used to establish an ensemble model for RUL prediction. In Reference 16, DBN is integrated with a particle filter and applied for RUL prediction of hybrid ceramic bearings.

2.2 | CNN

2D deep convolution neural network is proposed for RUL estimation based on time series from multi sensor signals splitting through sliding windows. Similarly in Reference 2, the convolution filters with 1-dimensional and different length are used. In Reference 22, CNN is used to obtain the eigenvector and a smoothing method linearly smooth the current forecast data to alleviate the problem of discontinuous predicted RUL.

AS time-domain and frequency-domain features can be only effective for a certain defect at a certain stage. The time-frequency domain information through short-time Fourier transform (STFT) is fed into CNNs. In Reference 19, a novel image feature-based CNN method with continuous wavelet transform (CWT) is applied for HI construction and RUL prediction. To consider the trend burr problem, a CNN-based HI construction method is proposed, where a new HI assessment metric is proposed to quantitatively evaluate the similarity of range scales of HIs in the same dataset. With the preprocessing of the Hilbert-Huang transform (HHT), a nonlinear degradation indicator is used as the label for learning feature of CNN model. And a support vector regression model is to predict RULs.

2.3 | RNN/LSTM

LSTM use input gate, forget gate, and output gate to control the information flow, therefore the long-term time dependency problems existing in RNN is solved. It is popularly applied for RUL estimation.

In Reference 26, a new LSTM network architecture is proposed, where a objective function and target RUL generation approach about the actual degradation of the system are introduced. By combining handcrafted feature design with automatic feature learning, a local feature-based gated recurrent unit (LFGRU) network approach is designed for machine health monitoring. These features extracted from each local window include time, frequency, and time-frequency feature. In Reference 25, a single layer perceptron with ReLU activation function is formulated to automatically derive the dimensionless health index from sensing data. Then a bi-directional LSTM is employed to track the variation of health index.

In Reference 12, the stacked sparse auto encoder automatically extracts degradation features from multiple sensors and logistic regression is used to predict the RUL. A LSTM-based encoder-decoder model is proposed to obtain a health index (HI) from multi-sensor time-series data, where the reconstruction error is used to compute HI. Similarly, in another work, RNN Encoder-Decoder (RNN-ED) is used. In Reference 30, the bidirectional RNN architecture on RNN-ED is proposed to learn more robust embeddings for constructing the one dimensional HI values. These methods do not rely on any degradation trend assumption based on domain knowledge.

To make RNN model be pay much attention to the significant information, gated dual attention unit (GDAU) neuro-network integrated with RMS health index is proposed for RUL eastimation.
2.4  |  Hybrid

In Reference 38, abstract features are learnt through RBM in the unsupervised pre-training stage and fed into LSTM to train with a supervised RUL regression way.

Because CNN and LSTM networks both possess unique abilities to learn features from data, it is reasonable to combine CNN with LSTM. In Reference 4, Convolutional Bi-directional Long Short-Term Memory networks (CBLSTM) are designed for RUL prediction. Similarly, 1D temporal convolutions are used to learn features relevant to time dependencies of sensor values. Then these extracted features are fed to a stacked LSTM network to learn the long short-term time dependencies.

2.5  |  Others

Using the pooling operation in CNNs will loss important information related with RUL, so CapsNets model is designed for RUL estimation. 39

To minimize data, the distribution discrepancy of the feature space between sample and target domain, transferable convolution neural network (TCNN) and Generative adversarial networks (GAN) are proposed to predict the RUL. 40, 41

3  |  OUR PROPOSED METHOD

The propose TCN method mainly includes three components in Figure 1, which are sequence data generation, TCN, and prediction module. Sequence data generation with sliding window is used to generate temporal sequence data from multi-sensors. TCN uses 1D convolution network, where causal filter and dilate convolution are used to expand the receptive field length. Finally, Dense Layer with ReLU activation function work as regression layer to predict the RUL value.

**FIGURE 1** The scheme of TCN for RUL prediction: sequence data from sensors are generated through sliding window and fed into temporal convolution network for RUL prediction, where causal filter and dilate convolution are used.
3.1 | Sequence data generation

To generate multi-variate temporal sequence data for training, a time window is adopted for time sequence processing shown in Figure 2. Let W denote the size of the time window. At each time step, all the past sensor data within the time window are collected to form a high-dimensional feature vector, and used as the inputs for the network. Therefore, each sample containing the temporal sequence information within the time window of W length. “m” is the number of sensors. “t” is current time. X in Equation (3) is all the temporal sequence time. Before the data are fed into the neuron network, they are normalized through Equation (4).

\[ x_t = [x_1^t, x_2^t, \ldots, x_m^t]^T \]  
\[ X^t = [x_{t-W+1}, x_{t-W+2}, \ldots, x_t] \]  
\[ X = [X^{W+1}, X^{W+2}, \ldots, X^{W+n}] \]  
\[ x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \]  

3.2 | The principle of TCN

3.2.1 | Sequence modeling

Given a sequence of input \( X = [x_0, x_1, \ldots, x_T] \), its corresponding sequence of output = \( [y_0, y_1, \ldots, y_T] \). And only those inputs that have been previously observed \( [x_0, x_1, \ldots, x_t] \) is used to predict the output \( y_t \). The prediction function can be expressed with Equation (5):

\[ \hat{y}_0, \hat{y}_1, \ldots, \hat{y}_T = f(x_0, x_1, \ldots, x_T) \]  

The goal of learning in the sequence modeling is to find the network \( f \) minimizing the expected loss between the actual outputs and predictions as Equation (6).

\[ L[(y_0, \ldots, y_T), (\hat{y}_0, \ldots, \hat{y}_T)] = L[(y_0, \ldots, y_T), f(x_0, \ldots, x_T)] \]  

The TCN produces an output of the same length as the input, so it uses a 1D fully convolutional network architecture, where each hidden layer is the same length as the input layer, and zero padding of length (kernel size—1) is added to keep subsequent layers the same length as previous ones.

3.2.2 | Causal convolutions

To achieve that there is no information “leakage” between future and past, the TCN uses causal convolutions, where an output at time \( t \) is convolved only with elements from time \( t \) and earlier in the previous layer in Figure 3.
Because models with causal convolutions do not have recurrent connections, they are typically faster to train than RNNs, especially when applied to very long sequences.

In Figure 3, the receptive field length $L_r$ is calculated as Equation (7).

$$L_r = 1 + (N_l - 1) \times (S_k - 1)$$  \hspace{1cm} (7)

where $N_l$ is the layer length, $S_k$ is the kernel size. According Equation (7), in order to achieve a long effective history size (receptive field length $L_r$), an extremely deep network $N_l$ or very large filter sizes $S_k$ is required. Therefore, one of the problems of causal convolutions is that they require many layers, or large filters to increase the receptive field.\textsuperscript{38} To resolve the problem, dilated convolutions (or convolution with holes) in Figure 4, where the filter is applied over an area larger than its length by skipping input values with a certain step, is used to increase the receptive field by orders of magnitude, without greatly increasing computational cost.\textsuperscript{36,38}

In Figure 4, $d$ is the dilation factor and dilation can be equivalent to introducing a fixed step between every two adjacent filter taps. The receptive field length $L_r$ is calculated as Equation (8).

$$L_r = \sum_d (S_k - 1) d$$  \hspace{1cm} (8)

where $D$ is the dilation array($d$). According Equation (8), choosing larger filter sizes $S_k$ or increasing the dilation factor $d$ can have long effective history size.

In this work, it is supposed that $d$ is increasing exponentially with the depth of the network, that is, $d = O(2^i)$ at level $i$ of the network. So Equation (9) can be deducted.
By comparing Equations (7) and (9), it can be concluded that dilation causal convolution have long effective history size. In common sense, it is easily accepted that the size of time window is just as the receptive field length. Therefore, in this work, the default size of time widow is the same with the receptive field length, so it can be calculated according Equation (9).

### 3.2.3 Residual block

According to Equation (9), increasing the depth of the network can increase the receptive field length. But it will lead to network performance getting saturated or degrading rapidly because of the notorious vanishing gradient problem, so ResNet introducing identity shortcut connection that skips one or more layers is used to ensure the stabilization of deeper and larger TCNs.

In the network model, residual block shown in Figure 5 is the basic unit, which effectively allows layers to learn modifications to the identity mapping rather than the entire transformation as Equation (10). And just as in Figure 6, the blocks are repeatedly shown, which benefit to very deep networks, TCN network.

\[
X_{\text{res}} = x + F(x, W)
\]  

### 3.3 Prediction layer

At last, the proposed architecture is implemented by combining temporal convolutional layers with regression layer, which use a fully connected layer with ReLU activation function.
4 | EXPERIMENTS

To evaluate the performance of the proposed method, two public datasets are used. In the experiments, the default dropout rate is 0.2. For simplicity, the default $W = 2^{N_l-1} \ast (S_k - 1)$. In addition, two evaluation metrics, scoring function and Root Mean Square Error (RMSE) in Equations (12) and (14), are used. The prediction error $d_i$ is the estimated RUL value $\hat{\text{RUL}}$ minus the ground truth value $\text{RUL}$.

$$d_i = \hat{\text{RUL}}_i - \text{RUL}_i$$ \hspace{1cm} (11)

$$\text{score} = \sum_{i=1}^{N} s_i$$ \hspace{1cm} (12)

$$s_i = \begin{cases} e^{-\frac{d_i}{\tau}} - 1 & d_i < 0 \\ e^{\frac{d_i}{\tau}} - 1 & d_i \geq 0 \end{cases}$$ \hspace{1cm} (13)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d_i^2}$$ \hspace{1cm} (14)

Accordingly, the curve of scoring function in Figure 7 penalizes late prediction (ie, the estimated RUL value is larger than the actual RUL value) more heavily than early prediction (ie, the estimated RUL value is smaller than the actual RUL value) since late prediction will lead to server fault event. In RUL prediction, it is reasonable that larger prediction errors are more severely penalized.
FIGURE 7 Evaluation metrics used in experiment: scoring function penalizes the late prediction more heavily

4.1 Turbofan engine degradation simulation dataset

Turbofan engine degradation simulation dataset in Table 1, C-MAPSS (Commercial Modular Aero-Propulsion System Simulation), is a widely used benchmark data. It has four sub-datasets with different number of operating conditions and fault conditions and each sub-dataset is further divided into training and test subsets. Each row in dataset has 26 columns, first column represents engine ID, second column represents the current operational cycle number, 3-5 columns are the three operational settings that have a substantial effect on engine performance, 6-26 columns represent the 21 sensor values.

Among 21 sensors, some sensors have constant output throughout the lifetime of the engine, which cannot provide any useful information to facilitate prediction. Therefore, the outputs of these sensors are removed from the C-MAPSS dataset as did in Reference 17. As a result, each data point involves 14 features corresponding to the outputs of these sensors with indices 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21. As FD002 and FD004 have six conditions, three operational settings are also chosen as the features.

In the training set of each sub-dataset, a linear or piecewise linear degradation model is used to estimate the ground-truth RUL value with respect to each training sample. Piecewise maximum RUL is 130 time cycles. In the test set of each sub-dataset, the actual ground-truth RUL value is available for each test sample.

In the experiment, except scoring function and RMSE, accuracy in Equation (16) is also used.

\[
TP_i = \begin{cases} 
1 & d_i \in [-13, 10] \\
0 & \text{others} 
\end{cases} \quad (15)
\]

\[
Acc = \frac{1}{N} \sum_{i=1}^{N} TP_i \quad (16)
\]

When the prediction error \(d_i\) of the \(i\)th data point falls within the range of \([-13, 10]\), the true positive \(TP\) is 1. The value of accuracy is the sum of \(TP\), which means that the total number of data points fall within the tolerance range.

| TABLE 1 C-MAPSS dataset |
|--------------------------|
| Dataset                  | FD001 | FD002 | FD003 | FD004 |
| Train trajectories       | 100   | 260   | 100   | 249   |
| Test trajectories        | 100   | 259   | 100   | 248   |
| Operating conditions     | 1     | 6     | 1     | 6     |
| Fault conditions         | 1     | 1     | 2     | 2     |
In a word, the reasons for choosing these three metrics are that larger prediction errors is not acceptable, late prediction will lead to server fault event and the tolerance to error is only acceptable within the specified range \([-13,10]\).

Through ten running with batch_size = 50, epochs = 500, and RMSProp optimizer used, the results are shown in Table 2. The proposed method outperforms other methods and achieves similar performance with CapsNet. CNN and Conv+LSTM obtain better results in RMSE values and RULCLIPPER shows its efficiency in score values.

Note that, in FD001, the RMSE value of TCN(14.9) is not lower than the RMSE values of BiLSTM-ED30(14.74) and RULCLIPPER38(13.27). But TCN gets the highest Accuracy(71), so lower RMSE cannot lead to the higher accuracy. Similarly, TCN gets lowest score value (203.04) and accuracy (71), but its RMSE value is not the highest. In FD003, TCN gets score value 838.18 and RMSE value 16.71, which are higher than the score value 317 and RMSE value 16 of RULCLIPPER.38 But TCN gets higher accuracy value 62, so results with lower score and RMSE cannot lead to the higher accuracy. Therefore, it is concluded that score and RMSE cannot evaluate the algorithm efficiency enough. Accuracy together with score and RMSE is better to evaluate the performance of algorithms.

In Figure 8, the prediction curve of RUL of FD003 is intensively fluctuated than others, which leads to the worst performance of score as shown in Table 2. It is because score is sensitive with big error according to Equation (13), especially for late prediction. After the fluctuation of curve in the beginning, the prediction RUL becomes smooth. Therefore, the tendency of RUL degradation is predicted more precisely with the time passed.

According to the defining of RMSE in Equation (14), it is sensitive with the fluctuation of RUL value, which forms the error value. Continue fluctuation of prediction value as shown in FD001 and FD003 leads to big RMSE in Table 2. However, accuracy can illustrate the number of error points falling in \([-13,10]\). The high accuracy means the perfect match between the predicted RUL and real RUL. Accuracy value is not sensitive with the small fluctuation of RUL.

In Figure 9, when the network level $N_l$ is increasing, the accuracy value is increasing and RMSE is decreasing. It is because the reception filed length is proportion with the depth of layer $N_l$ according to Equation (9). So does when the kernel size $S_k$ is increasing.

In Figure 10, when the network level is fix with $N_l = 4$, the accuracy value is also increasing and RMSE is decreasing. According to Equation (9), different kernel size can obtain different receptive filed length as the flow calculation.

\[
N_l = 4, S_k = 3 \text{ } L_r = 1 + (S_k - 1)(2^{N_l - 1} - 1) = 15.
\]

\[
N_l = 4, S_k = 5 \text{ } L_r = 1 + (S_k - 1)(2^{N_l - 1} - 1) = 29.
\]

\[
N_l = 4, S_k = 7 \text{ } L_r = 1 + (S_k - 1)(2^{N_l - 1} - 1) = 43.
\]

| Algorithms  | FD001 Score | Acc | RMSE | FD002 Score | Acc | RMSE | FD003 Score | Acc | RMSE | FD004 Score | Acc | RMSE |
|-------------|-------------|-----|------|-------------|-----|------|-------------|-----|------|-------------|-----|------|
| Proposed TCN | 203.04      | 71  | 14.90| 848.52      | 71  | 15.19| 838.18      | 62  | 16.71| 2259.20     | 57  | 19.74|
| CapsNet     | 276.34      | —   | 12.58| 1229.72     | —   | 16.30| 283.81      | —   | 11.71| 2625.64     | —   | 18.96|
| CNN²        | 274         | —   | 12.61| 10412       | —   | 22.36| 284         | —   | 12.64| 12466       | —   | 23.31|
| CNN¹⁸       | 1287        | —   | 18.45| 13570       | —   | 30.29| 1596        | —   | 19.82| 7886.4      | —   | 29.16|
| LSTM²³      | 338         | —   | 16.14| 4450        | —   | 24.49| 852         | —   | 16.18| 5550        | —   | 28.17|
| BiLSTM-ED³⁰ | 273         | 57  | 14.74| 3099        | 49  | 22.07| 574         | 42  | 17.48| 3202        | 40  | 23.49|
| Conv + LSTM³¹ | 1220    | —   | 23.57| 3100        | —   | 20.45| 1300        | —   | 21.17| 4000        | —   | 21.03|
| RULCLIPPER³⁸ | 216        | 67  | 13.27| 2796        | 46  | 22.89| 317         | 59  | 16.00| 3132        | 45  | 24.33|
| MODBNE¹⁷    | 334         | —   | 15.04| 5585        | —   | 25.05| 421         | —   | 12.51| 6557        | —   | 28.66|
| RF¹⁷        | —           | —   | 17.91| —           | —   | 29.59| —           | —   | 20.27| —           | —   | 31.12|
| GB¹⁷        | —           | —   | 15.67| —           | —   | 29.09| —           | —   | 16.84| —           | —   | 29.01|
| RVR³⁸       | 1503        | —   | 23.80| 17423       | —   | 31.30| 1432        | —   | 22.37| 26509       | —   | 34.34|

Table 2: Comparison

Note: RF, random forest; GB, gradient boosting; TCN: $W = \text{default}$, $N_l=5$, $S_k = 8$, number of filters = 55.
All these three lengths are less than $W = [50,75,100]$. So it can be concluded that with wide network width can improve the TCN performance.

### 4.2 Milling dataset

The publicly available milling dataset provided by UC Berkeley represent experiments conducted on a milling machine under various operating conditions. Flank wear is measured and used as an indicator of the health status of the milling machine. The data in Table 3 have three operating parameters: depth of cut (1.5 mm and 0.75 mm), feed rates (0.5 mm/rev and 0.25 mm/rev), and material type (cast iron and steel), and six different sensors: AC spindle motor current, DC spindle motor current, vibration at table, vibration at spindle, acoustic emission at table, and acoustic emission at spindle. There are 16 cases with varying number of runs for each case in the dataset. As Case #6 has only one run, it will be removed. The length of each sensor reading in a run is 9000.

In our experiment, when the flank wear is more than 0.45 for the first time, the RUL is set as 0, which is consistent with other publications. It was observed that there are obvious unstable regions in the initial and late stages due to the cut in and cut out processes. Only the readings in the middle stable region 3000-6000 (3000 values), down-sampling the original sensor readings by 30, were used to train.

To evaluate the performance of TCN method, mean absolute percentage error (MAPE) is used. But it is noted that MAPE will be failed when RUL$_i = 0$. 

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{d_i}{\text{RUL}_i} \right|$$  (17)
**FIGURE 9** Results with different parameters of FD001 ($N_L = [3, 4, 5], W = 2^{N_L-1} \times (S_k - 1)$, dilation=$N_L$)

**FIGURE 10** Results with different parameters of FD001 ($N_L = 4$)
### Table 3: Milling dataset

| Material      | #Case-number of runs-times | Depth of cut | Feed |
|---------------|----------------------------|--------------|------|
| 1-Cast iron   | #1-17-48                   | 1.5          | 0.5  |
| 1-Cast iron   | #2-14-72                   | 0.75         | 0.5  |
| 1-Cast iron   | #3-14-81                   | 0.75         | 0.25 |
| 1-Cast iron   | #4-7-39                    | 1.5          | 0.25 |
| 2-Steel       | #5-6-15                    | 1.5          | 0.5  |
| 2-Steel       | #6-1-0                     | 1.5          | 0.25 |
| 2-Steel       | #7-8-21                    | 0.75         | 0.25 |
| 2-Steel       | #8-6-12                    | 0.75         | 0.5  |

Since number of cases is small, leave one out method is used for model learning and parameters selection: 150 runs (15 case * 10 runs) for testify with batch_size = 500, epochs = 750, and Adam optimizer. The results are shown in Table 4 and Figure 11, where Cases #7 and #8 are not count on the average results and the results in the comma are obtained with the parameters $N_i = 5$, $S_k = 7$, $N_f = 30$, $W$ = default. As a result of RMSE in Table 4 shows that TCN outperforms than other methods and it does not perform well when steel material is as test set. It is because the time to collect data is

### Table 4: Comparison with milling dataset

| Algorithms | Cast iron | Steel |
|------------|-----------|-------|
|            | RMSE      | MAPE  | RMSE  | MAPE  |
| BiLSTM-ED30| 7.14      | 0.24  | 2.36  | 0.38  |
| LSTM-Recon28| 8.45    | 0.26  | 2.66  | 0.3   |
| TCN Iron#1 | 4.62      | 0.39  | 1.47  | 0.56  |
| TCN Iron#2 | 9.12      | 0.33  | ---   | ---   |
| TCN Iron#3 | 6.38      | 0.33  | 8.43(3.10) | 1.15(0.42) |
| TCN Iron#4 | 4.83      | 0.63  | 6.39  | 2.11  |
| TCN Iron#9 | 3.92      | 0.43  | 4.62  | 0.82  |
| TCN Iron#10| 3.56      | 0.32  | 1.18  | 0.36  |
| TCN Iron#12| 6.28      | 0.32  | 2.20  | 0.48  |
| TCNAvg(Above case) | 5.86   | 0.46  | 2.37  | 0.52  |

Note: TCN: $W = 200 N_i=4, S_k = 8, N_f = 30$, batch size is 500 and epoch number is 750.

![Figure 11](image-url) Results of RMSE and MAPE with milling dataset
small when steel material is used. For example, given in Table 3, most time used in steel cases is smaller than 20 seconds while most of iron cases is more than 40 seconds.

Figure 11 shows that Cases #2 and #8 obtained the worst results. One reason is that, paper\textsuperscript{30} points that one of the biggest challenges of the milling machine experimental data is that the degradation data is not strictly time-continuous as shown in Figure 12. The prediction value of Case #2 in Figure 13 between 0 and 400 is seriously deviated from the ground-truth value. This phenomenon is similar with the one of previous experiment, where the predicted RUL value in the beginning is intensively fluctuated. Therefore, the accuracy of the predicted RUL value in early stage of TCN is

![Graph showing degradation data for Cast Iron Case 2 and Steel Case 8](image)

**Figure 12** Raw signal (between 3000 and 6000) of Cases #2 and #8: the degradation data are not strictly time-continuous. For example, the data between 12,500 and 12,600 are zero.
one main problem. As for Case #8, one reason is that the time collecting data is only 22 seconds. Another reason is the degradation data is not strictly time-continuous shown in Figure 12.

5 | CONCLUSION

It is known that CNN can auto extract the spatial feature related with the dependencies of sensor values but it cannot extract the temporal feature related with time dependencies. Therefore, many methods combining CNN with LSTM are proposed for RUL prediction. Because change of RUL is a slow activity, it is better that long data history is to be learned in LSTM. However, LSTM has less efficiency in processing long history sequence. In this work, by combining causal filters with dilated convolutions, TCN is designed to expand the receptive field length which is suitable for domains where a long history is required. Through the experiments with two public dataset C-MAPSS and Milling, the proposed method can outperform most classic methods and achieve similar performance in C-MAPSS with CapsNet. TCN normally only requires three key parameters to be manually settled, which is easily applied in engineering. In contrast, the CapsNet network structure is required to be redesigned in different application fields.

Just as other methods do, it does not perform well in milling with steel dataset. One of the solution is to preprocess the raw dataset. As Reference 30 the linear interpolation strategy is used to reduce the gap between two consecutive runs of an instance into one cycle and different feature sets are tried according to their correlation. In addition, intensive fluctuation of RUL values at the beginning lead to the worst results of the score and RMSE, which is another shortage of TCN method. One solution to solve the problem is to use the rectified RUL value as Reference 2 Furthermore, late prediction related with the fluctuation of RUL should be attention in real engineering because it will lead to server accident with device damage. One solution is carefully evaluating the model with three evaluation metrics together. Better results of evaluation metrics are preferred because scoring function penalizes it more heavily and accuracy ensures the tolerance range of prediction error. Another solution is multi-stage identification method as paper45 proposed. As it said, a complete RUL decrease may consist of several degradation behavior patterns, in which dynamics keeps consistent in the same pattern, while is dissimilar between different patterns.

ACKNOWLEDGEMENTS
This work was supported in part by National Natural Science Foundation of China (61502423), Zhejiang Provincial Natural Science Foundation (LY16G020012), and Zhejiang Province Public Welfare Technology Application Research (LGF19F010002, LGF20F010004, Application Id GG21F030032:The Research on Cold-Start Fault Diagnosis Method Based on Immune Algorithm and Deep Learning).

PEER REVIEW INFORMATION
Engineering Reports thanks the anonymous reviewers for their contribution to the peer review of this work.
CONFLICT OF INTEREST
The authors declare no potential conflict of interest.

AUTHOR CONTRIBUTIONS
Jungan Chen: Formal analysis; investigation; methodology; project administration; software; validation; visualization; writing-original draft. Danjiang Chen: Formal analysis; methodology; software; validation. Gaoping Liu: Funding acquisition; resources; supervision; writing-review and editing.

DATA AVAILABILITY STATEMENT
The “Turbofan Engine Degradation Simulation Data Set C-MAPSS” data that support the findings of this study are openly available in the NASA Ames Prognostics Data Repository at https://ti.arc.nasa.gov/c/6/, reference number [42]. The “Milling Data Set” data that support the findings of this study are openly available in the NASA Ames Prognostics Data Repository at https://ti.arc.nasa.gov/c/4/, reference number [44].

ORCID
Jungan Chen https://orcid.org/0000-0002-7500-6031

REFERENCES
1. Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: a systematic review from data acquisition to rul prediction. Mech Syst Signal Process. 2018;104:799-834.
2. Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf. 2018;172:1-11.
3. Baraldi P, Mangili F, Zio E. Investigation of uncertainty treatment capability of modelbased and data-driven prognostic methods using simulated data. Reliab Eng Syst Saf. 2013;112:94-108.
4. Zhao R, Yan R, Wang J, Mao K. Learning to monitor machine health with convolutional bi-directional lstm networks. Sensors. 2017;17(2):273.
5. Si X-S, Wang W, Hu C-H, Zhou D-H. Remaining useful life estimation--a review on the statistical data driven approaches. Eur J Oper Res. 2011;213(1):1-14.
6. Wen J, Gao H, Zhang J. Bearing remaining useful life prediction based on a nonlinear wiener process model. Shock Vib. 2018;2018:1-13.
7. Lei Y, Li N, Gontarz S, Lin J, Radkowski S, Dybala J. A model-based method for remaining useful life prediction of machinery. IEEE Trans Reliab. 2016;65(3):1314-1326.
8. Li N, Lei Y, Yan T, Li N, Han T. A wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability. IEEE Trans Indus Electron. 2019;66(3):2092-2101.
9. Nguyen KTP, Medjaher K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliab Eng Syst Saf. 2019;188:251-262.
10. Ahmad W, Khan SA, Kim J. A hybrid prognostics technique for rolling element bearings using adaptive predictive models. IEEE Trans Indus Electron. 2018;65(2):1577-1584.
11. X. Jia, Bin Huang, Jianshe Feng, Haoshu Cai, and Jay Lee. A Review of PHM Data Competitions from 2008 to 2017: Methodologies and Analytics. Annual Conference of the PHM Society. 2018.10(1):1–10.
12. Ma J, Su H, Zhao WL, Liu B. Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning. Complexity. 2018;7(2018):1-13.
13. Deutsch J He D. Using deep learning based approaches for bearing remaining useful life prediction. Paper presented at: Annual Conference of the Prognostics and Health Management Society; 2016.
14. Liao L, Jin W, Pavel R. Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment. IEEE Trans Indus Electron. 2016;63(11):7076-7083.
15. Deutsch J, He D. Using deep learning-based approach to predict remaining useful life of rotating components. IEEE Trans Syst Man Cybernet Syst. 2018;48(1):11-20.
16. Deutsch J, He M, He D. Remaining useful life prediction of hybrid ceramic bearings using an integrated deep learning and particle filter approach. Appl Sci. 2017;7(7):649.
17. Zhang C, Lim P, Kai Qin A, Tan KC. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Trans Neural Netw Learn Syst. 2017;28(10):2306-2318.
18. Babu GS, Zhao P, Li XL. Deep convolutional neural network based regression approach for estimation of remaining useful life. International Conference on Database Systems for Advanced Applications. Lecture Notes in Computer Science; 2016;9642:214-228.
19. Yoo Y, Baek J-G. A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network. Appl Sci. 2018;8(7):1102.
20. Guo L, Lei Y, Li N, Yan T, Li N. Machinery health indicator construction based on convolutional neural networks considering trend burr. Neurocomputing. 2018;292:142-150.
21. Cheng C et al., A deep learning-based remaining useful life prediction approach for bearings. IEEE/ASME Trans Mechatron. 2020;25(3):1243-1254.
22. Ren L, Sun Y, Wang H, Zhang L. Prediction of bearing remaining useful life with deep convolution neural network. IEEE Access. 2018;6:13041-13049.
23. Zheng S, Ristovski K, Farahat A, Gupta C. Long short-term memory network for remaining useful life estimation. Paper presented at: 2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017; June 19–21, 2017; Dallas, TX: IEEE:88-95.
24. Yuting Wu MY, Dong S, Lin L, Liu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing. 2018;275:167-179.
25. Zhang J, Wang P, Yan R, Gao RX. Long short-term memory for machine remaining life prediction. J Manuf Syst. 2018;48:78-86.
26. Elsheikh A, Yacout S, Ouali M-S. Bidirectional handshaking LSTM for remaining useful life prediction. Neurocomputing. 2019;323:148-156.
27. Zhao R, Wang D, Yan R, Mao K, Shen F, Wang J. Machine health monitoring using local feature-based gated recurrent unit networks. IEEE Trans Indus Electron. 2018;65(2):1539-1548.
28. Malhotra P, Vishnu TV, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G. Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder. 2016. https://arxiv.org/pdf/1608.06154.pdf.
29. Gugulothu N, Vishnu TV, Malhotra P, Vig L, Agarwal P, Shroff G. Predicting remaining useful life using time series embeddings based on recurrent neural networks. 2017. https://arxiv.org/abs/1709.01073.
30. Yu W, Kim IY, Mechefske C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mech Syst Signal Process. 2019;129:764-780.
31. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. Mech Syst Signal Process. 2019;115:213-237.
32. Li X, Zhang W, Ding Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. Reliab Eng Syst Saf. 2019;182:208-218.
33. Jayasinghe LS, Samarasinghe T, Yuen C, Ge SS. Temporal convolutional memory networks for remaining useful life estimation of industrial machinery. 2019 IEEE International Conference on Industrial Technology (ICIT). 2019:915-920.
34. Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. 2018. https://arxiv.org/abs/1803.01271.
35. Oord AV, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. WaveNet: a generative model for raw audio. 2016. https://arxiv.org/abs/1609.03499v2.
36. Bai S, Kolter JZ, Koltun V. Convolutional sequence modeling revisited. Paper presented at: 6th International Conference on Learning Representations; 2018. https://openreview.net/forum?id=rk8wKk-R-.
37. Qin Y, Chen D, Xiang S, Zhu C. Gated dual attention unit neural networks for remaining useful life prediction of rolling bearings. Reliab Eng Syst Saf. 2019;185:372-382.
38. Ellefsen AL, Bjørlykhaug E, Esøy V, Ushakov S, Zhang H. Remaining useful life predictions for turbofan engine degradation using semisupervised deep architecture. Reliab Eng Syst Saf. 2019;183:240-251. https://doi.org/10.1016/j.ress.2018.11.027.
39. Ruíz-Tagle Palazuelos A, Drogueut EL, Pascual R. A novel deep capsule neural network for remaining useful life estimation. Proc Inst Mech Eng J Risk Reliab. 2020;234(1):1-17.
40. Cheng H, Kong X, Chen G, Wang Q, Wang R. Transferable convolutional neural network based remaining useful life prediction of bearing under multiple failure behaviors. Measurement. 2021;168(1):1-31.
41. Li X, Zhang W, Ma H, Luo Z, Li X. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. Knowl Based Syst. 2020;197:105843.
42. Ramasso E, Saxena A. Performance benchmarking and analysis of prognostic methods for cmapss datasets. Int J Progn Health Manag Forum. 2014;5(2):1-15.
43. Ramasso E. Investigating computational geometry for failure prognostics. Int J Progn Health Manag. 2014;5:1-18. https://hal.archives-ouvertes.fr/hal-01303405.
44. Agogino A, Goebel K. Milling Dataset, BEST Lab, UC Berkeley, NASA Ames Progn. Data Repos; 2007. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.
45. Qin Y, Yuen C, Adams S. Invariant learning based multi-stage identification for lithium-ion battery performance degradation; 2020. https://arxiv.org/abs/2008.05123.