ISA Transactions 123 (2022) 200–217

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

Causal augmented ConvNet: A temporal memory dilated convolution model for long-sequence time series prediction

Abiodun Ayodeji [a], Zhiyu Wang [a], Wenhai Wang [a], Weizhong Qin [b], Chunhua Yang [c], Shenghu Xu [b], Xinggao Liu [a,*]

[a] State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, PR China
[b] China Petroleum Chemical Co. Jiujiang Branch, Jiujiang 332004, PR China
[c] School of Information Science & Engineering, Central South University, Changsha 410083, PR China

ARTICLE INFO

Article history:
Received 31 December 2020
Received in revised form 15 May 2021
Accepted 16 May 2021
Available online 19 May 2021

Keywords:
Dilated convolution neural network
Deep learning
Remaining useful life
Time series
Predictive maintenance

ABSTRACT

A number of deep learning models have been proposed to capture the inherent information in multivariate time series signals. However, most of the existing models are suboptimal, especially for long-sequence time series prediction tasks. This work presents a causal augmented convolution network (CaConvNet) and its application for long-sequence time series prediction. First, the model utilizes dilated convolution with enlarged receptive fields to enhance global feature extraction in time series. Secondly, to effectively capture the long-term dependency and to further extract multiscale features that represent different operating conditions, the model is augmented with a long-short term memory network. Thirdly, the CaConvNet is further optimized with a dynamic hyperparameter search algorithm to reduce uncertainties and the cost of manual hyperparameter selection. Finally, the model is extensively evaluated on a predictive maintenance task using the turbofan aircraft engine run-to-failure prognostic benchmark dataset (C-MAPSS). The performance of the proposed CaConvNet is also compared with four conventional deep learning models and seven different state-of-the-art predictive models. The evaluation metrics show that the proposed CaConvNet outperforms other models in most of the prognostic tasks. Moreover, a comprehensive ablation study is performed to provide insights into the contribution of each sub-structure of the CaConvNet model to the observed performance. The results of the ablation study as well as the performance improvement of CaConvNet are discussed in this paper.

1. Introduction

A time-series signal is a sequence of data usually acquired successively at a uniformly spaced interval over a specified period. Conventional time series sequence contains temporal, recurring patterns with an autocorrelated structure that inherently defines the process from which they are acquired. Time series signals also contain critical historical and current information about the performance, dynamics and operating characteristics of processes, systems, and structures. The information in time series variables are being applied to solve different problems across various fields. In heavy industries, the inherent patterns in time series signals are used to provide insights into the reliability, sustainability, and maintainability of critical components, tools and devices. Characteristic patterns are also prominent in time series signals acquired from smart systems in manufacturing industries, which informed their use in real-time process status updates, fault diagnosis, prognosis, and predictive maintenance tasks.

Time series signals can be univariate or multivariate. A univariate time series contains a single signal varying with time while a multivariate time series consists of multiple observations that are simultaneously changing over time. The patterns in multivariate time series signals have been used to predict process variable dynamics in chemical plants [1], to improve electric valve reliability [2], to study the thermal-hydraulic processes in nuclear plants [3], to predict the wear in spur gears [4], to dynamically monitor gear performance online [5], and to analyze industrial IoT equipment working condition [6]. Moreover, the advent of big data, ensured by improved data acquisition methods results in the surge in the novel application of data-driven models. For instance, shallow models like support vector machine [7–11], neural networks [12–16], and other hybridized probabilistic and evolutionary optimization approaches [17,18] have been used to predict the reliability of submarine engines [18], for multiplicative fault diagnosis [19], to predict heat exchanger tube defects [20], for ship propulsion system health monitoring [21],

* Corresponding author.
E-mail address: lxg@zju.edu.cn (X. Liu).

https://doi.org/10.1016/j.isatra.2021.05.026
0019-0578/

to monitor electric valve health [22], and for vehicle remaining useful life estimation [23]. However, most of these models are sub-optimal either because the model performance depends on extensive feature extraction and feature selection expertise (as in shallow machine learning models) or the model output has significant uncertainty (as in probabilistic models).

Recently, the automatic feature extraction capability of deep learning approaches is being utilized to extract features in multivariate time series measurements. This results from attempts to address the inefficiency in conventional data-driven networks that require hand-coded features [24]. Deep learning models have been widely used for prognostic and health management tasks such as gear remaining useful life prediction [25], and induction motor rolling element bearing failure prediction [26]. One of the most prominent multivariate benchmark datasets used in evaluating predictive deep learning algorithms is the NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset. The dataset generated from the simulation of dynamic, nonlinear processes is being used to develop predictive models with the capability to learn the features in the previously observed signals to predict the remaining useful life (RUL) of turbofan aircraft engines. Considering the importance of the C-MAPSS dataset to this work, it is pertinent to discuss the state of the art in the utilization of the benchmark dataset.

The long-term sequences in the CMAPSS dataset have been used to verify the predictive capability of a stochastic model [27]. A Gated recurrent unit (GRU) based neural network has also been used to extract and learn the nonlinear degradation patterns in the C-MAPPS dataset [28]. The approach adopts a two-stage system comprising a data dimensionality reduction and the GRU model, and a comparison with LSTM shows a better performance for the proposed GRU. Ellefsen et al. [29] also proposed an unsupervised pre-trained model for turbofan RUL prediction. The work utilized an unsupervised and semi-supervised approach to train a model to extract the features in the C-MAPSS dataset and then used the genetic algorithm to improve the model's prediction. A dual LSTM with change detection capability has also been presented [30], where a health index construction function is used to aid the recurrent network in detecting change points in sensor measurements.

The recurrent networks presented above are useful in understanding the patterns in the CMAPSS dataset. However, they are sub-optimal. Consequently, the convolution neural network (CNN) and the generative auto-encoders are some of the most widely-used recent deep learning models for RUL prediction. At its core, CNNs are used to capture spatially invariant features in images and patterns in text. The network can learn the patterns efficiently, and it has been proven to perform well in image classification tasks [31]. Many CNN-based deep learning models have also been proposed to capture patterns and structures in the C-MAPSS dataset. A common approach to obtain an optimal model is integrating the models as a hybrid. Some implementation also stacks conventional CNN with recurrent or fully connected layers to improve model accuracy [32]. For instance, Berghout et al. [33] proposed a technique that uses a denoising auto-encoder for feature extraction, and a dynamic forgetting factor to retain recent training data points during the training of an extreme learning machine. The data filtering capability of the method and the pre-training feature extracting by the denoising autoencoder gives the model improved sensitivity to the degradation patterns in the dataset. The evaluation result is reported to have shown a stable network response even under random solutions. An online–offline training approach applied to bidirectional LSTM with the encoder–decoder model has also been used to predict the remaining useful life of turbofan engines [34]. To ensure generalization, Al-Dulaimiet al. [35] integrated two deep learning models in parallel to estimate the RUL of the turbofan engine. The approach combines LSTM with CNN to extract both temporary and spatial features, and the output of the hybrid model is fused to estimate the RUL. Table 1 summarizes some of the recent deep learning models used to learn the degradation pattern in the C-MAPSS dataset.

Table 1
Recent deep learning models that use CMAPSS dataset for RUL prediction (the years between 2018–2020).

Author and Refs	Year	Approach
Ellefsen et al. [5]	2020	RBM+LSTM
Shi et al. [6]	2020	Dual-LSTM
Berghout et al. [7]	2020	OS-ELM
Chen et al. [4]	2019	GRUNN
Al-Dulaimi [10]	2019	HDNN
Wu et al. [33]	2019	DLSTM
Wang et al. [8]	2018	BiLSTM
Li et al. [9]	2018	DCNN

The state-of-the-art methods presented in Table 1 offer different comparative advantages. However, they are sub-optimal when applied to capture the long-term sequence and time-varying operating conditions inherent in heavy industrial equipment's run-to-failure dataset. Moreover, in most of the previous work studied, the hyper-parameters are not dynamically optimized, giving room for subjective, hand-coded, and costly hyperparameter selection processes which render most of the models unrepeatable for different prognostic tasks. Besides, some of the presented models suffer from exploding trainable parameters resulting in model complexity [36].

A promising application of the CNN has been presented as WaveNet [37]. Inspired by the WaveNet architecture, this work proposes a causal, augmented convolution network (CaConvNet) with the capability to capture the temporal pattern as well as the long-term sequence in dynamic long-sequence time series datasets. The proposed CaConvNet uses dilated convolution to increase the receptive field exponentially as a function of the convolution layer. The carefully selected dilation rate of the model is augmented with the memory retention capability of a long-short-term memory network to capture temporal patterns and local dependencies inherent in multivariate signals. This results in a model that detects fine details by processing inputs in higher resolutions, capture more contextual information, and trains faster. Specifically, this work offers the following contributions:

(i) A new causal dilated convolution network with enlarged receptive fields is proposed for robust time series prediction.
(ii) To effectively capture the long-term dependency inherent in a run-to-failure dataset, the proposed approach is augmented with a long-short term memory network.
(iii) To effectively address the manual and highly subjective hyperparameter selection involved in deep learning model development, the proposed CaConvNet is optimized with a dynamic hyperparameter selection algorithm. The production-level optimization approach utilized also supports task-specific application, ensures speedy convergence and reduce the risk of overfitting.
(iv) The proposed model is extensively evaluated using four different datasets that show the degradation pattern of turbofan aircraft engines. The model's performance is also evaluated against seven state-of-the-art models previously used for RUL prediction. The CaConvNet evaluation results show significant performance improvement over the state-of-the-art models.

The sections below describe the development of the proposed CaConvNet, and the experiments performed to evaluate the predictive performance of the model. The data preparation and detailed modeling techniques to achieve effective prediction are presented in Section 3. The model cross-validated training approach, and special callback functions implemented to reduce the risk of overfitting are also presented in Section 3. The CaConvNet model is evaluated on the C-MAPSS dataset in Section 4, using similar metrics common in recent turbofan engine RUL prediction work. Section 5 summarizes and concludes the study.

2. The proposed CaConvNet method

The proposed CaConvNet utilizes a multi-layer, dilated convolution network with enlarged receptive fields, and a temporal memory network as its core. The CaConvNet model is inspired by the architecture of the WaveNet model. Hence, to properly define the CaConvNet model, it is necessary to give a brief background on the WaveNet model. The waveNet model contains a series of stacked one-dimensional dilated convolutions with an exponentially increasing dilation rate. The causal convolution ensures that the future sequences are not used to predict the past, allowing the model to conserve the ordering in the dataset. For instance, a model prediction $p\left(k_{t+1}|k_1,\ldots,k_T\right)$ on the ground truth k does not depend on future time steps $k_{t+2}, k_{t+3},\ldots,k_T$ [34]. This is achieved by shifting the output of a normal convolution by a few timesteps, implemented by padding the input with zeros. However, in WaveNet architecture, the paddings are all added in the front ('causal' padding). This architecture performs quite well at text-to-speech tasks. However, the padding and the structure of the convolution layer in the conventional WaveNet architecture is not optimal in time series prediction task that has long-term dependencies, and unknown or randomly varying future conditions. This limitation makes the conventional waveNet unsuitable to capture the dynamics inherent in some time series prediction tasks. Moreover, although the causal padding can properly handle the temporal flow, it is well suited for time-series prediction where the dependencies are short-term and for a small number of time steps. In addition, some time-series tasks have spatial dependencies across a large number of time-steps, which requires a well-suited, specialized model. The temporal position matters in most time series prediction tasks that have long-term dependencies because different interpretations are required for new and old data points.

To address the identified issues, this work proposes CaConvNet with modified architecture. For the proposed CaConvNet model, the front-only padding is not utilized because, during training, the conditional prediction for all timesteps can be made in parallel if all the timesteps of the ground truth are known. However, for some predictive tasks, some of the timesteps for the ground truth are unknown, as in the case of the CMAPSS test set. Also, because a one-dimensional (1D) convolution layer process input patches independently, the front-only causal padding is not sensitive to the order of the timesteps. Hence, in the proposed convolution network, padding is added on both sides of the input so that the size of each feature map in the layer's output is equal to the size of each input feature map. For a one-dimensional convolution with input x, size N, filter size M_1, the kth layer output feature map o^k can be expressed as [38]:

$$o^k(i,h) = \left(w_h^k * x\right)(i) = \sum_{j=-\infty}^{\infty} w_h^k(j)\, x(i-j) \tag{1}$$

where $o^k \in \mathbb{R}^{k\times N-l+1\times M_1}$, w is layer k's filter, l is the filter size, and $h=1,\ldots,M_1$. Without zero paddings, the width of the convolution output for all k is $N_k = N_{k-1} - l + 1$, for $k = 1,\ldots,K$. Also, the shared weight characteristics of all elements in a feature map allow the model to detect time-invariant features. For a padded input with a vector of leading zeros of the size of the receptive field, the input size that controls the convolution output is given by:

$$[0,\ldots,0,x(0),\ldots,x(N-1)] \in \mathbb{R}^{N+r} \tag{2}$$

While for a padded input vector with leading-and-trailing zeros of the size of the receptive field implemented in the proposed CaConvNet, the input size that controls the network output is given by:

$$[0,x(0),\ldots,x(N-1),0] \in \mathbb{R}^{N+r} \tag{3}$$

where r is the receptive field, and the layers of zeros can be adjusted appropriately according to the filter size to compensate for the border effect. The causal interpretation is retained in the proposed method by the training approach used. Thus, the receptive field of the network contains $x(0)$ *and*$x(t)$ when predicting $x(t+1)$. The training approach ensures that the time step given as the target value comes after the final layer's receptive field. Hence, the model never has access to future time steps when predicting the value of the current one.

Fig. 1(a) and (b) shows the fundamental differences in the convolution layers between the proposed method and the conventional WaveNet architecture. Apart from the causal convolution structure of waveNet, the model architecture utilizes other techniques such as residual connection, skip connection, and a gated recurrent unit, as shown in the box portion of Fig. 1(a). For tasks where lower-level signals are useful for prediction, the skip connection is used to preserve the information as the input propagates across the network hierarchy. Similarly, the residual connection ensures the collected layer-wise output is combined for final processing. For some function f, that represents the model learned weight, the residual connection ensures the network output is mapped to the input as:

$$x_{out} = f(x_{in}) + x_{in} \tag{4}$$

As opposed to the traditional $x_{out} = f(x_{in})$. Residual connection facilitates the use of a deeper network by allowing for more direct gradient flow in backpropagation [39]. These techniques make significant differences in a task involving 3-dimensional inputs and hundreds or thousands of layers seen in image recognition tasks. In the proposed method, these techniques are not utilized because of the relatively small number of layers used. Moreover, when the conventional WaveNet is evaluated with the multivariate time series run-to-failure dataset, it produces suboptimal models. Moreover, the proposed architecture in Fig. 1(b) shows sparsity introduced by utilizing the *ReLU* activation unit, expressed as:

$$ReLu(x) = \max(0, a * x) \tag{5}$$

Eq. (5) above defines the required input configuration. The function is zero for all negative values of x, and a^*x otherwise, where a is a learnable parameter. In this implementation, the sparsity gained by introducing *ReLu* addresses the vanishing gradient problem, and the "*dead ReLU*" problem does not occur because all input from a run-to-failure dataset is positive. The advantage of *ReLu*'s sparsity and fast convergence is utilized in the subsequent layers in the CaConvNet, described in Section 3 below. The theory and impact of other CaConvNet optimization techniques are presented in Sections 2.1 and 2.2.

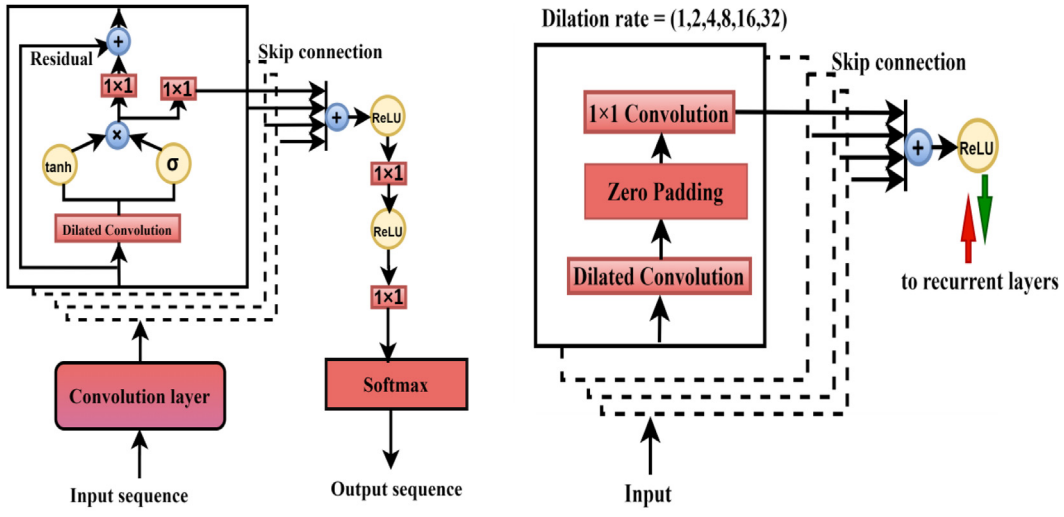

Fig. 1. (a) Conventional WaveNet architecture. (b) CaConvNet' sub-structure.

2.1. Receptive field enlargement with dilated convolution

Recently, several novel applications of the convolution network have been proposed, including different utilization in benchmark studies [40,41]. This is because the size and number of time-steps in a typical multivariate run-to-failure dataset makes it impractical to use only the conventional convolution neural network. Also, padded convolutions have the risk of network explosion because of extra neurons added at each end of the network. To handle model complexity, the proposed CaConvNet uses dilated convolution to increase the receptive field exponentially as a function of the convolution layer. In a 2D or 3D convolution neural network, the receptive field is the region of the input space that influences the network output feature. To compute the effective receptive field, consider a convolutional neural network with an input signal f_0, L number of layers, (where $\{L|1 < l \leq L\}$), and final output f_L, the output of the lth layer with a height h_l, width w_l and depth d_l is given by:

$$f_l \in \mathbb{R}^{h_l \times w_l \times d_l} \tag{6}$$

In each layer L, the spatial configuration is parameterized by the kernel size k_l, the left-side padding p_l, the right-side padding q_l, and the stride s_l. Hence, the receptive field size r_l of the final output feature f_L is defined as [42]:

$$r_l = s_{l+1} \cdot r_{l+1} + (k_{l+1} - s_{l+1}) \tag{7}$$

Eq. (7) can be re-written as:

$$r_{l-1} = s_l \cdot r_l + (k_l - s_l) \tag{8}$$

Eq. (8), referred to as the recurrence equation, can be solved by multiplying it by $\prod_{i=1}^{l-1} s_i$ to have:

$$r_{l-1} \prod_{i=1}^{l-1} s_i = r_l \prod_{i=1}^{l-1} s_i + k_l \prod_{i=1}^{l-1} s_i - \prod_{i=1}^{l-1} s_i \tag{9}$$

If $A_l = r_l \prod_{i=1}^{l-1} s_i$, and $\prod_{i=1}^{0} s_i = 1$, then $A_0 = r_0$, and Eq. (9) can be re-written as:

$$A_l - A_{l-1} = \prod_{i=1}^{l-1} s_i - k_l \prod_{i=1}^{l-1} s_i \tag{10}$$

Summing Eq. (10) from $l = 1toL$, Eq. (10) becomes:

$$\sum_{l=1}^{L} A_l - A_{l-1} = A_L - A_0 = \sum_{l=1}^{L} \prod_{i=1}^{l-1} s_i - k_l \prod_{i=1}^{l-1} s_i \tag{11}$$

Since $A_0 = r_0$, and $A_L = r_L \prod_{i=1}^{L} s_i$, then the full convolution network receptive field is computed as:

$$r_0 = \prod_{i=1}^{L} s_i + \sum_{l=1}^{L} \left(k_l \prod_{i=1}^{l-1} s_i - \prod_{i=1}^{l} s_i \right) \tag{12}$$

Swapping variables in Eq. (12), the final expression for the receptive field size r_0 is given as:

$$r_0 = \sum_{l=1}^{L} \left((k_l - 1) \prod_{i=1}^{l-1} s_i \right) + 1 \tag{13}$$

The receptive field is characterized by its location and size [42]. For a one-dimensional dataset, the objective is to maximize the size of the field of the input signal that contributes to the output features by increasing the network receptive field. This is to ensure that both the spatial and temporal information inherent in the degradation dataset is effectively captured. This approach has been found to increase model accuracy significantly [42]. Hence, the goal is to design a causal convolution network with a receptive field that covers the entire relevant input space.

Dilation is conventionally used to enlarge the receptive field, to integrate information from different spatial scales, and to balance both the local accuracy as well as the global knowledge without parameter explosion. To effectively increase the receptive field, a dilation rate is introduced. Given a sequence of input $x \in \mathbb{R}^n$, and filter $f : \{0, \ldots .k-1\} \rightarrow \mathbb{R}$, dilation introduces a "hole" in the convolution without changing its weights [40]. Dilating a kernel by a factor of α introduces striding of α between the samples used when computing the convolution. That is, the span of the kernel (k>0) is increased to $\alpha\,(k-1)+1$ as the receptive field size is increased. In contrast to the conventional convolution, the receptive field size increment of one-dimensional convolution operation F of sequence s is defined as:

$$F\,(s) = r_0 = \sum_{l=1}^{L} \left(\alpha(k-1)_l \prod_{i=1}^{l-1} s_i \right) + 1 \tag{14}$$

A network of multi-layer dilated convolution is constructed with linearly increasing dilated factor with layer depth. The dilation rate also ensures that the filters skip the processed input at a constant rate, as opposed to directly applying filters to the inputs sequentially. This lets the convolutional layer have a larger receptive field at no computational price and using no extra parameters. The larger the receptive field, the better the network, up to a certain threshold. This characteristic is useful for large time series with thousands of time steps. The output of each layer after the convolution operation can be described as:

$$c_i = \sigma\left(\sum_{l=1}^{L}\left(w^l\right)^T x^l_{i:\,i+k-1} + b^l\right) \tag{15}$$

where σ is the restricted linear activation unit, b^l is the bias term for the lth feature map, k is the kernel size, i is the number of units, and w^l is the weight for feature map l. The dilated convolution is used to filter the input. Following the convolution layer is the pooling layer. The convolution and pooling operations are sequential, with the output of the convolution acting as the input of the pooling layer. The pooling layer extracts useful information by aggregating and transforming the local features. Considering the stacks of convolution layers used in the model, pooling is useful to significantly reduce the number of trainable parameters while extracting critical features in the time series. The maximum value selection in the pooling layer can be described as:

$$p^l_i = \max_{r\in R}\left(c^l_{i\times T+r}\right) \tag{16}$$

where T is the stride, R is the size of the pooling layer, and p is a one-dimensional single layer vector of the form $p=[p_1,\ldots,p_l]$. In summary, the leading-and-trailing padding ensures the limitations identified for models with only leading padding are avoided. The causal, dilated convolution is implemented to ensure the network trainable parameters does not explode, considering the input sequence. The output of the pooling layer is subsequently used as the input into the LSTM layer described below.

2.2. Causal dilated convolution network augmentation with LSTM

In heavy industries, many components and systems are connected to form the required process workflow. These interconnections also influence how components in the system fails. For instance, the interaction between components with different operating characteristics may result in common cause failure, influence the time to failure of critical systems, or result in catastrophic domino effect. Also, external conditions may accelerate a component's degradation or failure, and neglecting such scenario may result in an inaccurate estimation of the safety and reliability of the component. Moreover, in some studies that utilized the CMAPSS dataset for model evaluation, a form of filter is always applied to reduce "noise". However, noise is a common characteristic of real-world industrial systems. Besides, the supposed noise may contain inherent information that describes the operating environment of the component. A robust model must be able to learn these characteristics without costly hand-coded features. Consequently, in the development of CaConvNet, recurrent memory cells that enables the model to decouple and extract multiscale features from high-dimensional, noisy data under time-varying component operation is introduced.

Long-short term memory network (LSTM) is a type of recurrent neural network with the capability to learn and retain order and temporal patterns in sequences with thousands of timesteps. As shown in the computation graph for the memory portion of an LSTM (Fig. 2) [43], a typical LSTM cell contains the forget gate, the candidate layer, the input gate, the output gate, the hidden state, and a memory state. For a given input vector $x_{(t)}$, the mathematical formulation of LSTM units comprising the input gate $x_{(t)}$ the forget gate $f_{(t)}$, the output gate $o_{(t)}$, a new memory cell $\bar{c}_{(t)}$, the final memory cell $c_{(t)}$, and the current cell output $h_{(t)}$ is expressed as:

$$i_{(t)} = \sigma(W_{(i)}x_{(t)} + U_{(i)}h_{(t-i)}) \tag{17}$$

$$f_{(t)} = \sigma(W_{(f)}x_{(t)} + U_{(f)}h_{(t-i)}) \tag{18}$$

$$o_{(t)} = \sigma(W_{(o)}x_{(t)} + U_{(o)}h_{(t-i)}) \tag{19}$$

$$\bar{c}_{(t)} = tanh(W_{(c)}x_{(t)} + U_{(c)}h_{(t-i)}) \tag{20}$$

$$c_{(t)} = f_{(t)} * \bar{c}_{(t-1)} + i_{(t)} * \bar{c}_{(t)} \tag{21}$$

$$h_{(t)} = o_{(t)} * \tanh(c_{(t)}) \tag{22}$$

where $h_{(t-1)}$ is the previous cell output, $\bar{c}_{(t-1)}$ is the previous cell memory, and W, U are the weight vectors. The capability of LSTM to retain the long- and short-term memory in the cell state prevent gradients from vanishing. These characteristics of LSTM are utilized to augment the convolution layers in the model. This is because, beyond a local scale, the convolution layers are not sensitive to the orders of the timesteps.

In this work, the proposed CaConvNet utilizes the multi-layer convolution kernels to extract different features, which are then aggregated in a feature map through multiple convolutions. The dilated convolutions used as the preprocessing layers of the network down-sample the input sequences into shorter, higher-level features that can be easily learned by the recurrent part of the network. To recognize longer-term patterns, a two-layer LSTM is used to process the outputs from the dilated causal convolution. The LSTM layer processes the input, and its outputs are used by the fully connected layers for prediction. A detailed description of the proposed architecture is shown in Fig. 3, and its application to predicting the remaining useful life of turbofan engines is described in Section 3.

3. Proposed CaConvNet implementation for RUL prediction

3.1. Problem formulation

Predictive maintenance problem in industrial components involves making the high-level connection between time-to-failure and available data point. Taking reference from the individual unit (turbofan engine in this case), all historical sensor measurements $x(0)$, $x(1)$,…,$x(t)$ that represent input variables across multiple sensors need to be consolidated at the global and local level to predict future outputs $\hat{x}(t+1)$. The CMAPSS dataset used in this work has been benchmarked, with each variable acquired at the appropriate time step and frequency. To establish the robustness of the dataset, most of the engines are working under different time-varying operating conditions, which would require the additional task of simulating the engine's operation-specific deterioration and its future stress load. The problem is to estimate the time to failure, and the response variable is taken as the number of cycles remaining for the engine before it fails. In this work, a piecewise representation of the lifecycle of the engine is taken as the response variable. One of the data preprocessing tasks is to label the response variables that indicate the entire degradation profile, from the onset to failure. Hence, the task of predicting the remaining useful life or the probability of turbofan engine failure at a period in the future can be formulated as a regression problem that involves minimizing the model's mean squared error by tuning the network weights and obtaining a model with the capacity to learn the behavior of the acquired variable, and generalize appropriately to unseen sensor measurements. Model performance evaluation also requires a consolidated metric for a specific use case.

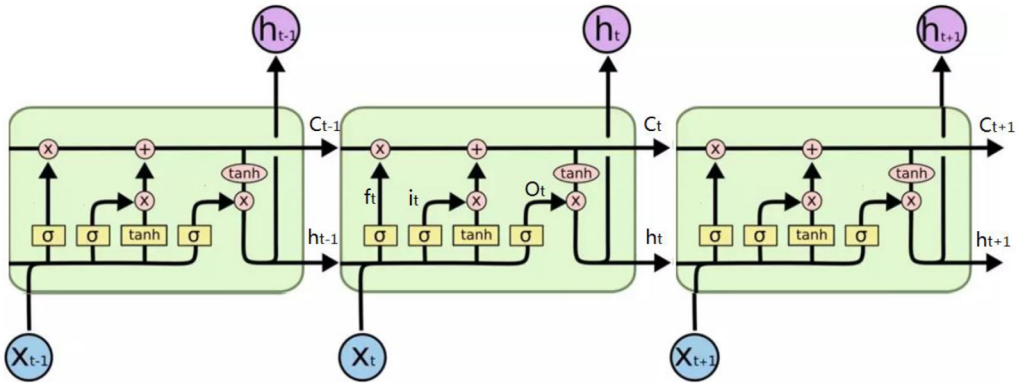

Fig. 2. The LSTM Computation graph.

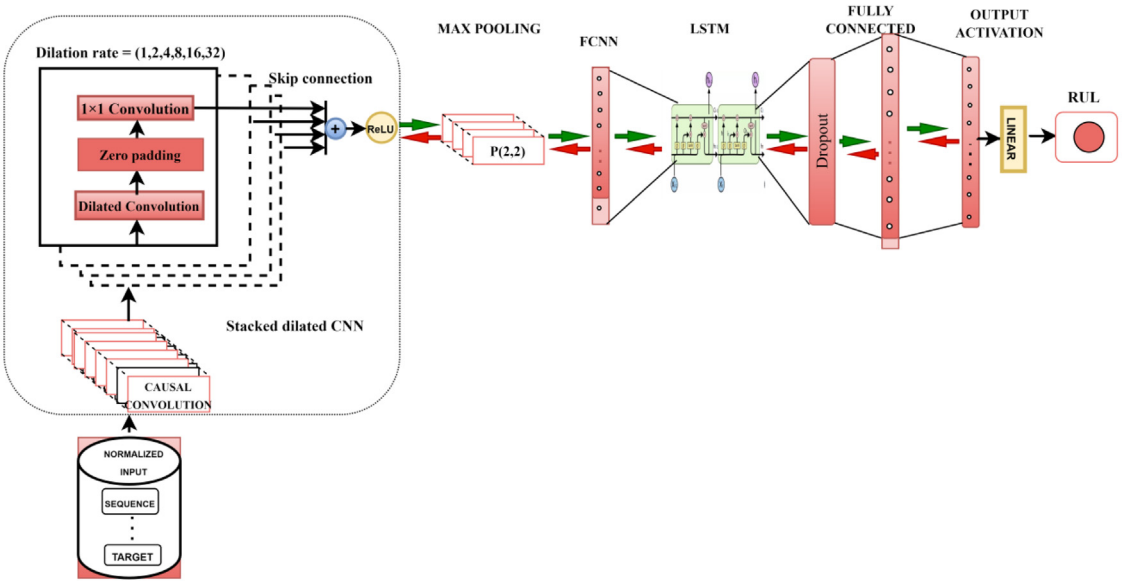

Fig. 3. Proposed CaConvNet architecture for remaining useful life prediction.

3.2. Experiment

In this work, the models are developed using Keras API with Tensorflow backend. The experiments presented in Section 4.3 are performed on an intel core i-7 workstation with RTX 2060s GPU. All other experiments are performed on Google Colab, an online cloud-based platform with accelerated compute environments, including GPUs and TPUs. Moreover, to facilitate reproducibility, an annotated jupyter notebook with codes and dependencies, and the trained models are provided in the first author's GitHub repository.

3.2.1. Dataset description

The proposed CaConvNet is evaluated on a Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset. The C-MAPSS dataset contains 4 sub-sets (FD001, FD002, FD003, FD004) that represents the sequences obtained from 21 sensors in a realistic simulation of the degradation of a large commercial turbofan aircraft engine provided by NASA [44]. The dataset is a record of series of engine flight conditions and different fault modes, with each sub-set containing one training set and one test set. The FD001 test set contains 13,092 measurements representing 100 different turbofan engines, FD002 has 33,992 measurements representing 259 engines. FD003 and FD004 have 16,597 and 41,215 with 100 and 248 engines respectively. The detailed description of the composition of the C-MAPSS dataset and the turbofan engine flight condition indicators are presented in Tables 2 and 3 respectively. In this study, an exhaustive evaluation of the proposed method is carried out on all four datasets.

Table 2
Composition of the C-MAPSS dataset.

Dataset	FD001	FD002	FD003	FD004
No of engines in the training set	100	260	100	249
No of engines in the test set	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2
Training size	20632	53760	24721	61250
Test size (default)	13097	33992	16597	41215

3.3. Data preprocessing

3.3.1. Sensor selection

Each subset of the C-MAPSS dataset contains 26 variables of which 21 are sensor readings. However, for subset FD001 and FD003, some sensor measurements do not provide additional information to aid the prognostic task [28,29]. Hence, some signals with the most informative features are selected. In this work, based on the performance of the signals selected in [29,30,33] for data subset FD001, sensors [1,5,6,10,16,18,19] and setting 3 are

Table 3
Turbofan engine condition indicators.

Variable no	Description	Units
1	Total temperature at fan inlet	R̊
2	Total temperature at LPC outlet	R̊
3	Total temperature at HPC outlet	R̊
4	Total temperature at LPT outlet	R̊
5	Pressure at fan inlet	Psia
6	Total pressure in bypass-duct	Psia
7	Total pressure at HPC outlet	Psia
8	Physical fan speed	Rpm
9	Physical core speed	Rpm
10	Engine Pressure ratio	–
11	Static pressure at HPC outlet	Psia
12	Ratio of fuel flow to Ps30	pps/psi
13	Corrected fan speed	Rpm
14	Corrected core speed	Rpm
15	Bypass ratio	–
16	Burner fuel–air ratio	–
17	Bleed enthalpy	–
18	Demanded fan speed	Rpm
19	Demanded corrected fan speed	Rpm
20	HPT coolant bleed	lbm/s
21	LPT coolant bleed	lbm/s

discarded. Also, for data subset FD003, sensors [1,5,16,18,19] and setting 3 are also discarded. All the datasets in the subset FD002 and FD004 are used for training.

3.3.2. Data normalization

For each subset in the C-MAPSS dataset, the selected signals are normalized before using it for training. The signals are transformed by scaling each feature with the Scikitlearn's *MinMax Scaler*. The *MinMaxScaler* rescales the dataset such that all sensor values are in the range [0, 1]. Subsequently, for computation optimization, a data generator is used to generate the sequences in the form [*Dataframe, sequence_length, sequence_column*], where the *Dataframe* is the preprocessed dataset, *sequence_length* is the selected time step (time window), and the *sequence_column* is the selected features for each data subset.

3.3.3. RUL target function

The task of estimating the remaining useful life involves determining the number of cycles remaining before the engine fails. Since the engine health is inversely proportional to the engine cycles, this task is achieved by creating an artificial signal that represents the number of cycles that remain before the engine failure. Meanwhile, previous observation on component degradation pattern shows that significant degradation is not common at the beginning of life. This observation motivated the augmentation of the beginning of life target output to a constant value of 130 (i.e. $R_{\text{early}} = 130$) for all sequences. This means that for the first 130 cycle, the engines function effectively before the onset of degradation afterwards. This target function augmentation, referred to as the piece-wise RUL is utilized to better represent the true output and obtain a reasonable estimation of the RUL for the turbofan engine.

3.3.4. Cross-validation effect

Considering data acquisition challenges, it is pertinent to adopt a training method that maximizes the utilization of the available dataset. The K-fold cross-validation approach is a principled model training approach that has been found to provide state-of-the-art result in recent benchmark studies [40]. Cross-validation is also useful to measure the performance of a model more accurately on new data points. For the *K-Fold* cross-validation approach, the training dataset are divided into k-folds of which (k-1) folds are used as training dataset and a fold is used as validation dataset at training time. This approach may introduce class imbalance in the train-validation fold, which would be more pronounced if the RUL estimation is presented as a classification problem. Consequently, *StratifiedKFold,* a variant of the KFold cross-validation approach, is used to prepare a once-through dataset after the k-fold splits. In the stratified approach, the class distribution in the dataset is preserved in the training and validation splits. For the stratified K-Fold cross-validation approach applied in this work, the model runs through the entire training set k times during training, and at each time, a different split is used for model validation. The test set in the CMAPSS dataset is separately preprocessed and used to test the model after training. The approach uses each non-overlapping part of the split as the validation set, by preserving a percentage of samples for each fold. Fig. 4 illustrates the *stratifiedKFold* approach as used in the model training.

Table 4
The hyperband algorithm.

The hyperband algorithm for hyperparameter optimization

Input : R, η $(default\ \eta = 3)$
initialization : $S_{max} = \lfloor log_{\eta}(R) \rfloor, B = S_{max} + 1)R$
for $S \in \{s_{max}, s_{max} - 1, \ldots, 0\}$ *do*
  $n = \lceil \frac{B}{R} \frac{\eta^*}{(s+1)} \rceil, \quad r = R\,\eta^{-s}$
  // begin SuccessiveHalving with (n,r) inner loop
  $T =$ *get_hyperparameter_configuration(n)*
  for $i \in \{0, \ldots, s\}$ *do*
    $n_i = \lfloor n\,\eta^{-i} \rfloor$
    $r_i = r\,\eta^{i}$
    $L = \{run\_then\_return\_loss(t, r_i) : t \in T\}$
    $T = top\_k(T, L, \lfloor n_i/\eta \rfloor)$
  end
end
return configuration with the smallest intermediate loss seen so far

3.3.5. Dynamic model hyperparameter selection

The hyperparameters are the variables that determine the performance of predictive models. A good hyperparameter is necessary to optimize model performance, as deep learning algorithms are sensitive to hyperparameter settings. However, selecting an optimal hyperparameter is an art, and the state-of-the-art deep learning models utilized to learn important information in CMAPSS relies on the subjective experience of the developer. To reduce the subjective characteristics of optimal hyperparameter selection, an automated tool is utilized for the task.

In this work, the Hyperband algorithm is used to dynamically select the optimal hyperparameters. The Hyperband tuning algorithm relies on adaptive resource allocation technique and early-stopping criterion to quickly estimate the parameters needed for a high-performing model. Its uniqueness is its ability to evaluate more configurations than other procedures, and its capability to adapt to unknown convergence rates. Table 4 shows the implementation of the hyperband algorithm for hyperparameter optimization. A detailed description of the algorithm and its application to other tasks can be found in [45].

A properly instantiated tuner functions with a well-defined search space. The hyperband search space is defined by specifying the CaConvNet as the hyper model, to be searched for 40 epochs over 20 trials, and the validation loss is selected as the objective to be minimized. Although the dynamic tuning capability provided by the tuner is desired, it also adds a significant computational burden. Hence, this work also presents an implementation of the proposed model with hyperparameters selected based on best-performing use-case on a similar task.

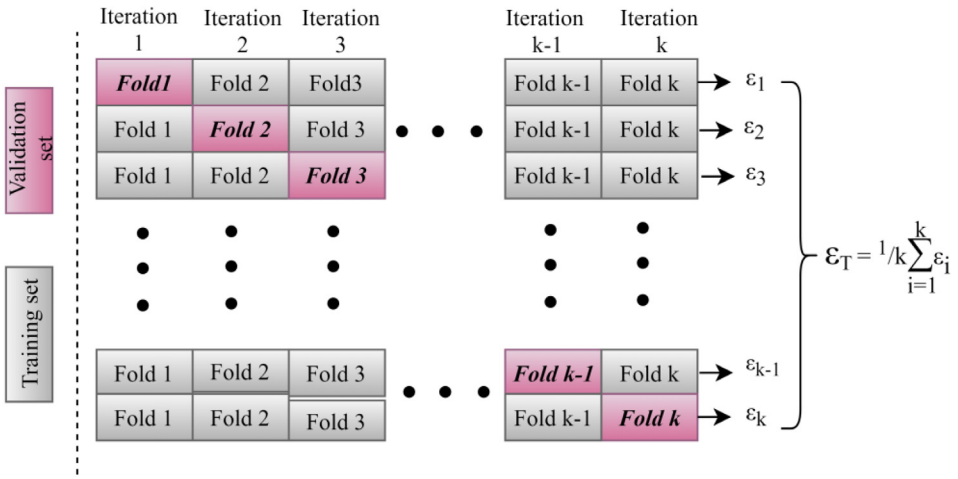

Fig. 4. Stratified K-fold cross-validation approach.

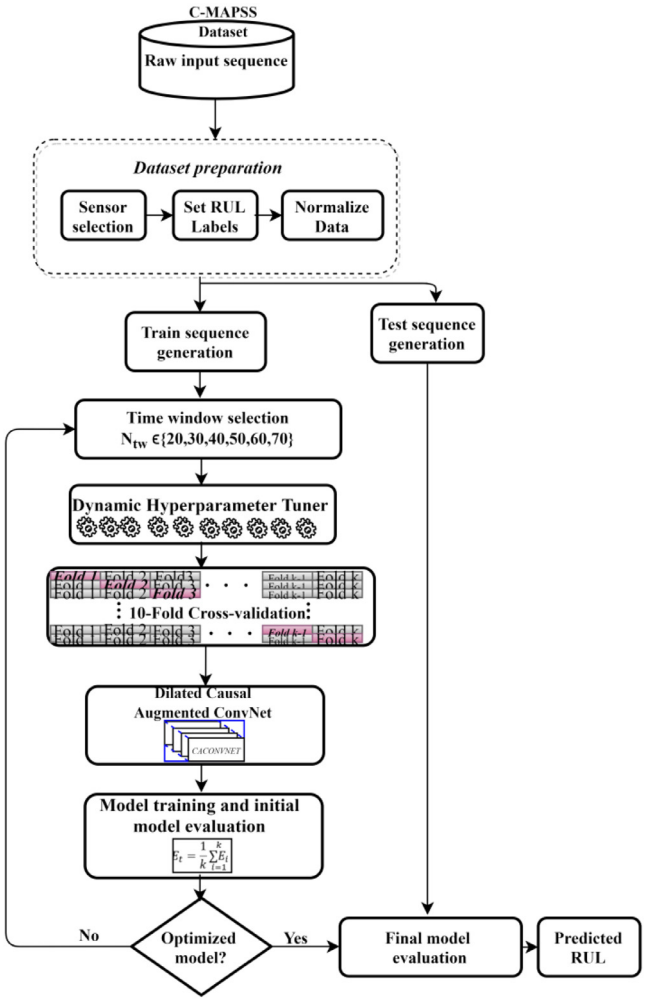

Fig. 5. The flowchart of RUL prediction with CaConvNet.

3.4. Performance evaluation metric

To ensure a fair comparison, this work uses the metrics commonly used to evaluate predictive models that utilized C-MAPSS dataset. The scoring metric and the root mean squared error (RMSE) has been widely used to estimate the relative performance of prognostic and health monitoring models. The scoring metric and the RMSE are used to rate the predictive model performance on the test dataset, and are defined as:

$$Score = \begin{cases} \sum_{i=1}^{n} \left(\exp\left(-\frac{e_i}{13}\right) - 1\right), & if\ e_i < 0 \\ \sum_{i=1}^{n} \left(\exp\left(-\frac{e_i}{10}\right) - 1\right), & if\ e_i \geq 0 \end{cases} \tag{23}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{24}$$

where e_i is the difference between the estimated RUL and the actual RUL ($RUL_{true} - RUL_{predicted}$) for the *ith* test unit. The task is to minimize the score and RMSE value such that late predictions ($e_i>0$) are more heavily penalized than early predictions ($e_i<0$). In either case, the RUL output that diverges from the ground truth is penalized exponentially, which makes the score increases exponentially with the prediction error. This is because late prediction risks catastrophic failure in critical components, while early prediction gives more time for maintenance actions to be performed on monitored component or system.

3.5. Model development and architecture

The 16-layer CaConvNet has six one-dimensional dilated convolution layers, two recurrent layers, and four fully connected connections. A predictable problem is that the network is likely to overfit the training data since it has many more adjustable weights. Hence, a max-pooling layer (MaxPool1D), and a dense layer (with a dropout rate of 0.2) is placed between the convolution and the LSTM layers. Moreover, two Keras callback functions – *EarlyStopping* and *ModelCheckpoint* are implemented. The *EarlyStopping* function checks at the end of every epoch if the metric to be optimized (loss) is no longer improving, then stops model training after a pre-defined epoch, while the *ModelCheckpoint* saves the best model according to the quantity monitored, which is later loaded for evaluation. To significantly speed up model training and reduce overhead, the Keras' *steps_per_execution* function is implemented during model compilation. This function specifies the number of training batches to process sequentially in a single execution. Table 5 presents some of the training parameters used in the model development, and Fig. 5 shows the proposed model architecture as applied for RUL prediction.

Table 5
Layer details and parameter settings for the proposed method.

Parameter	Value
Number of Epochs	500
Learning rate	0.00003
Optimizers	Adam
Loss function	Huber
Model checkpoint (save best only)	True
Early stopping/patience	True/10
Steps per execution	10
StratifiedKFold split	10
Dynamic tuner maximum trial	20
Execution per trial	2
Hyperband maximum epoch	40
Trainable parameters	463,333
Layer number	16
Hidden activation	ReLu
Output activation	Linear

4. Results and analysis

This section discusses the comprehensive evaluation result of the proposed CaConvNet for RUL estimation. To verify the CaConvNet parts that contribute to the improved model performance, an extensive ablation study of the model is also presented. Moreover, this section demonstrates the superiority of the proposed approach by comparing the result with WaveNet, conventional deep learning models, and other state-of-the-art prognostic results in the literature.

4.1. Prognostic performance of CaConvNet

Previous models evaluated on the C-MAPSS dataset are silent on the model performance on the full test subset of the C-MAPSS dataset. Also, the conventional approach is to select a few data points (last sequence for each engine) in the test dataset to evaluate the model. These evaluation approaches are weak, as the model performance on the full test spectrum as well as under various operating conditions is critical. Hence, in this work, the proposed CaConvNet is evaluated on the full test set, as shown in Figs. 6–9. The figures show the performance of each model on the respective test datasets.

Figs. 6–9 shows the accuracy of the model prediction on the out-of-sample test dataset. It is observed that the model prediction is consistent with the actual RUL for the engine, especially for FD001 and FD003 datasets. For FD002 and FD004, the dataset's diversity (each engine record under six different operating conditions) and noise in the dataset account for the little variation seen at the peaks and troughs of the RUL plots. However, the model's ability to learn in the presence of such diversity is displayed in the consistent trend shown at the end of the RUL plots as shown in Figs. 7 and 9.

To further analyze the proposed model and demonstrate its superiority. Fig. 10(a)–(d) shows the predictive performance of the model using selected engines. The engines are randomly selected to verify the RUL estimation performance of the model using representative samples. As seen in Fig. 10(a)–(h), the initial RUL prediction was a bit noisy towards the middle of engine life especially in Fig. 10(c), (d), and (g). However, as the engine's remaining useful life decreases, the model's RUL evaluation also improves, as shown at the tail end of the curve. This results from the fact that when the unit is close to its end of life, the degradation signatures are enhanced, enabling the proposed model to accurately capture the patterns for better prognostics. Moreover, the piecewise-linear approach of approximating the beginning of life for each engine proves to be a better way of framing the prognostic problem. This is evident in how the model's estimated RUL at the beginning of life is close to the engine's constant early life condition (R_{early}). Prediction for datasets FD002 and FD004 is not smooth at the middle of engine life because of the multiple fault mode represented in the dataset. Despite some deviation in the actual and predicted RUL of the model on some of the datasets, the prognostic accuracy is generally high especially when the units are close to failure. This is critical for prognostic and health management of industrial components, as accurate prediction at the engine's end of life enables prompt maintenance response.

4.2. Ablation study

4.2.1. Effect of different window length (N_{tw})

For a dataset with a sequential structure, the sequence length (or window length) is one of the most important parameters directly relevant to the ultimate accuracy of deep learning model used to capture the information in the dataset [28]. The window length is commonly used in sequence analysis tasks, with each window describing a batch of the dataset. Given a set of sequence S of length n defined as S = (s1,.....sn), with labels L specified as the required RUL, where si $\in L$ for all i$\in$ 1,.....n, the aim is to find a set Θ of k window lengths that are most informative considering how well the predicted RUL correlates with the ground truth. However, window length selection is often subjective, and to the best of the authors' knowledge, there is no earlier work with specific optimal window length selection criteria.

Hence, to find the most informative window for a particular subset of the C-MAPSS dataset, we experimented with a small set $\Theta \in$ (20, 30, 40, 50, 60, 70) of interesting windows that represent different lengths selected in recent literature. These window lengths are sequentially evaluated on each dataset using the proposed method and the results are as shown in Table 6. Table 6 shows that different time windows have different effect on different datasets. The table is a demonstration that the selection of accurate window length is critical to determine the appropriate levels of granularity for analyzing the underlying pattern in long-sequence time series data, and its relevance to the ultimate accuracy of deep learning models. For FD001- 4, the most effective time window on the proposed model is N_{tw} = 30, 60, 50, 50 respectively. It is also observed that the window N_{tw} = 50 performs relatively better on dataset FD001, and the result is close to the optimum output from N_{tw} = 30. Although the experiment is revealing, it comes with a certain computational burden. Hence, for off-the-shelve implementation of the model with a single-window length across the dataset, time window N_{tw} = 50 is recommended.

4.2.2. Effect of different model features

To better understand the causality in the model, and to further explore model parts that contribute to the improved performance, this section discusses the effect of the combination of different layers and parameters on the model. As shown in Table 5, the final CaConvNet model contains six (6) dilated convolution layers, one max-pooling layer, two (2) LSTM layers, and three (3) fully-connected layers (with two dropout layers that randomly select the input units with frequency 0.2 at each time step during training). To measure causality, the model is randomly initialized with different layer numbers and parameter values, and separate experiments are performed for each dataset. The ablation study result is as shown in Table 7.

The comparison of the 2-layer CaConvNet and 4-layer CaConvNet shows decreased performance for the 4-layer implementation compared with the 2-layer, except on the data subset FD003, where the 4-layer implementation has a 23.40% decrease in the score value and 12.19% decrease in the RMSE. Incidentally, there is a significant improvement in the prediction result

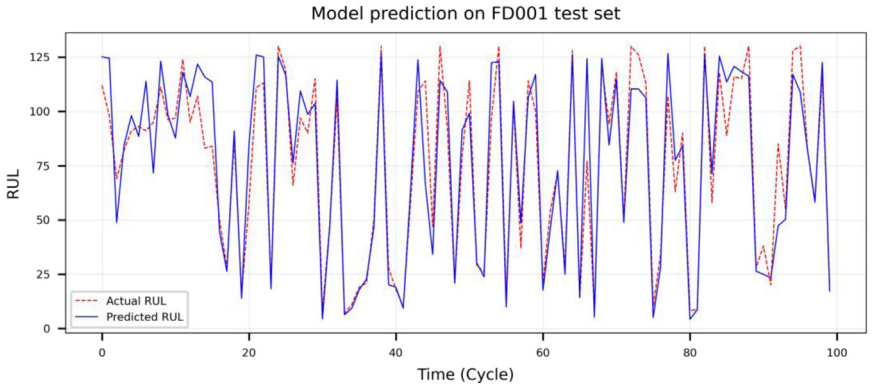

Fig. 6. CaConvNet prediction on FD001 test set.

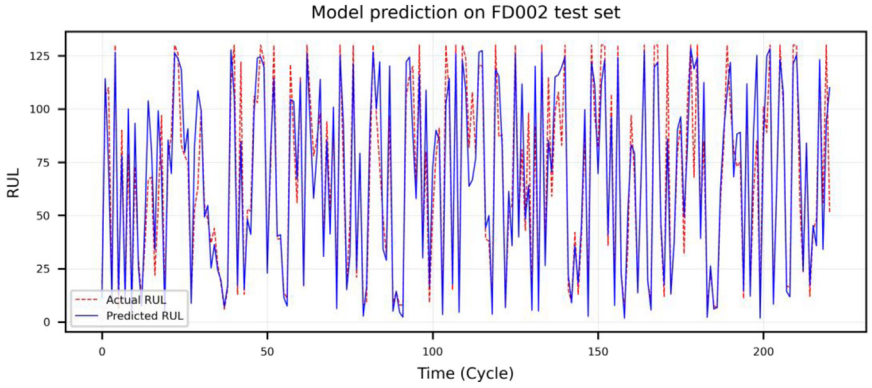

Fig. 7. CaConvNet prediction on FD002 test set.

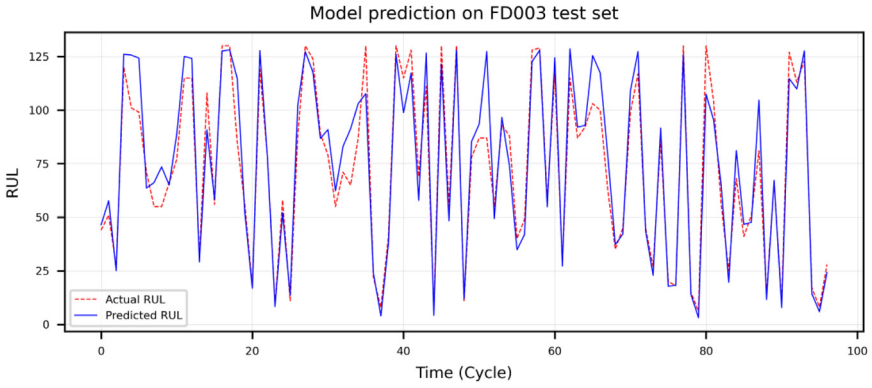

Fig. 8. CaConvNet prediction on FD003 test set.

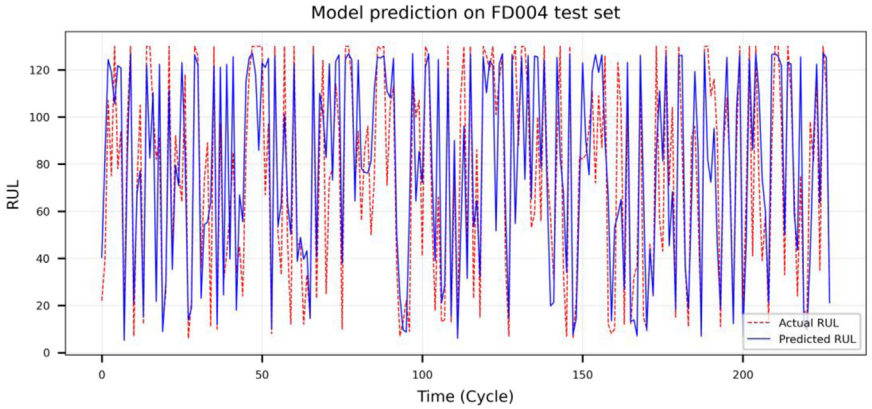

Fig. 9. CaConvNet prediction on FD004 test set.

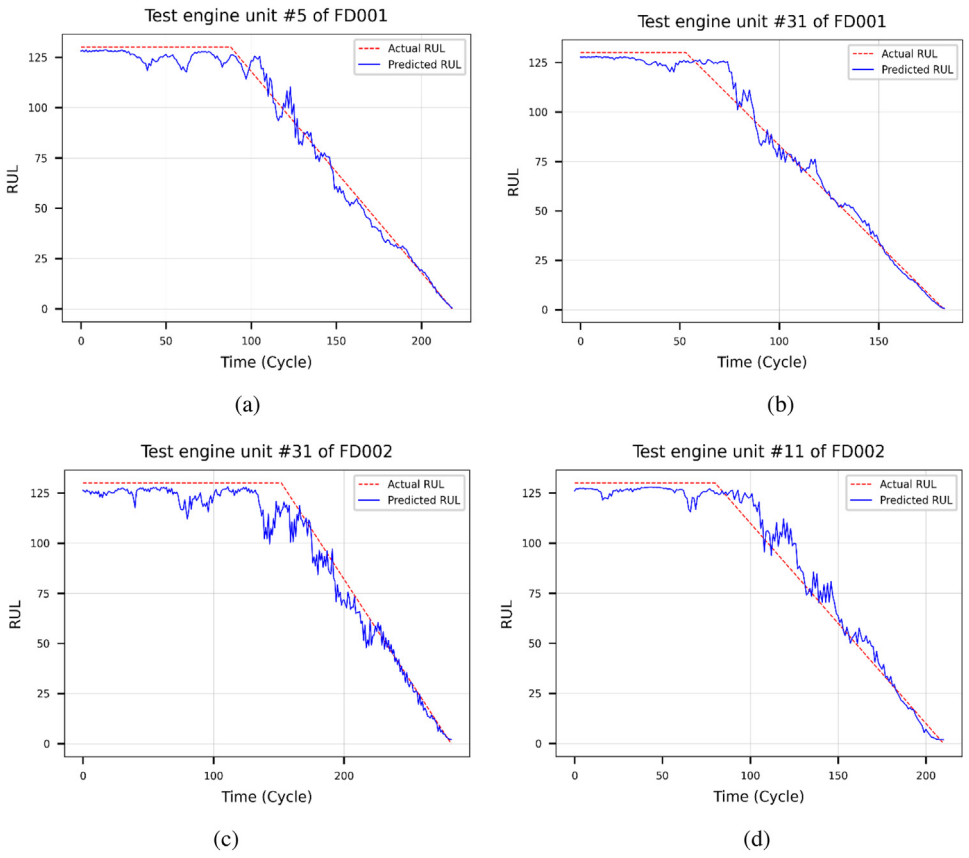

Fig. 10. (a) & (b) are the RUL prediction for engine unit no 5 and 31 in FD001, (c) & (d) are RUL prediction for engine unit no 11 and 31 in FD002; (e) & (f) are the RUL prediction for engine units 34 & 96 in FD003; and (g) &(h) are the predictions for engine units 1 and 37 in FD004 respectively.

Table 6
Effect of different window length (N_{tw}).

Dataset		$N_{tw} = 20$	$N_{tw} = 30$	$N_{tw} = 40$	$N_{tw} = 50$	$N_{tw} = 60$	$N_{tw} = 70$
FD001	Score	2777.74	**84.83**	136.86	85.18	120.23	10586.34
	RMSE	20.15	**11.83**	14.32	12.57	13.50	24.8421
FD002	Score	312.11	522.94	689.78	1459.57	**226.33**	640.71
	RMSE	20.94	21.82	22.50	22.29	**15.12**	22.30
FD003	Score	398.62	155.94	130.35	**63.64**	1750.18	96.55
	RMSE	14.82	12.25	12.39	**9.63**	13.53	11.06
FD004	Score	1014.25	999.78	1551.27	**1179.63**	7455.11	3291.36
	RMSE	25.57	25.54	29.04	**22.31**	36.23	26.98

Table 7
Ablation study of the proposed CaConvNet.

Model	FD001		FD002		FD003		FD004	
	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE
2-layer CaConvNet	101.19	12.43	235.08	15.67	104.265	12.39	3236.22	22.61
4-layer CaConvNet	114.62	13.43	952.47	23.24	79.87	10.88	4704.6	29.01
7-layer CaConvNet	8796.91	41.39	1873.72	inf	74.46	11.00	1194.05	23.48
CaConvNet without Dense	103.84	11.99	230.8	16.81	74.83	10.01	2125.78	25.03,
Undilated CaconvNet-with augmentation	102.36	12.48	137.59	15.58	97.25	10.90	1196.13	30.24
Dilated CaConvNet-no augmentation	10701.71	17.83	10929.89	19.02	62379.26	21.54	51805.77	25.70
CaConvNet without CV	178.87	13.02	13864.36	45.83	261.54	16.21	29102.33	48.83
CaConvNet with CV	**84.83**	**11.83**	**226.33**	**15.12**	**55.52**	**9.24**	**1179.63**	**22.31**

between the 4-layer model and the proposed 6-layer model, as shown in Table 7. To further confirm the convolution layer effect, we also experimented with a 7-layer CaConvNet. A significant decline in performance is observed, compared with the proposed 6-layer approach, across FD001- 4 dataset (score: −99.03%, −87.92%, −25.44%, −1.2%, and RMSE: −71.42%, inf., −15.99%, −4.98% on FD001-4 dataset respectively). For CaConvNet without the fully connected layers, the result shows a consistent decrease in performance compared with the CaConvNet implementation with the fully connected layers (score: 18.31%, 1.94%, 25.81% and 44.51%; RMSE: 1.33%, 10.05%, 7.69% and 10.87% improvement for FD001-4 respectively). Moreover, the effect of the receptive field enlargement was also evaluated. The result shows a significant improvement for the score metric (FD001:17.13%, FD002: 64.49%,

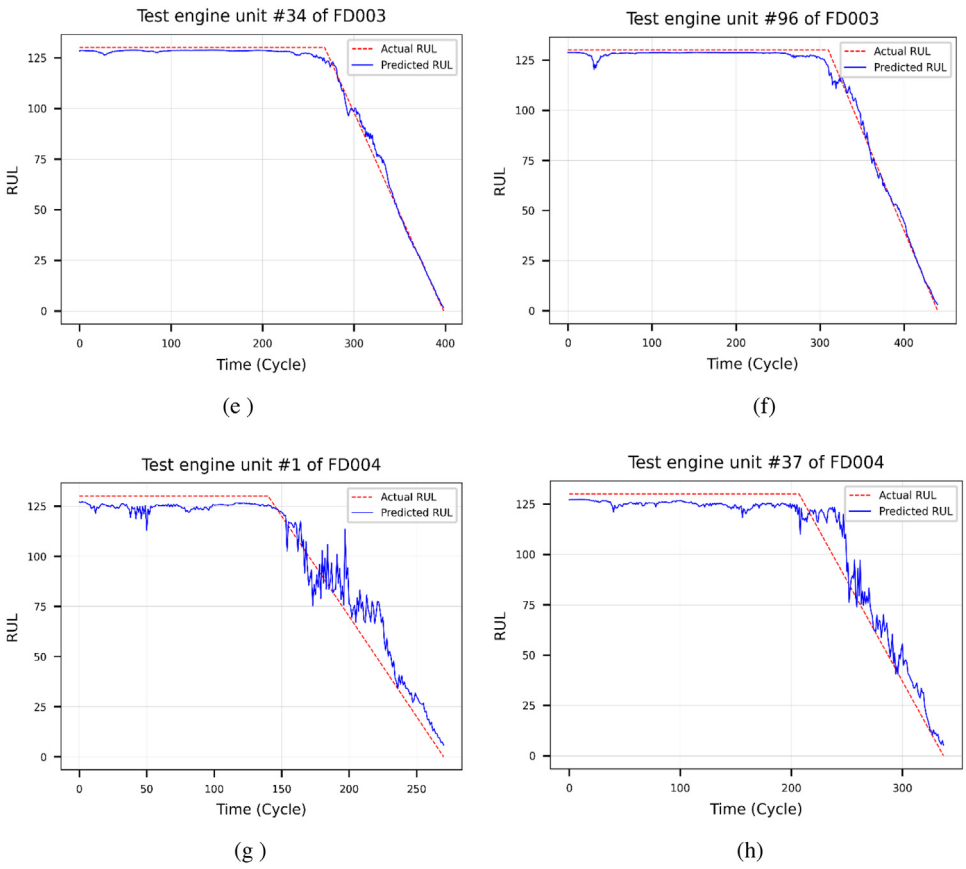

Fig. 10. (*continued*).

FD003: 42.91, and FD004:1.38%) and RMSE (FD001:5.21%, FD002: 2.95%, FD003: 15.23%, and FD004:26.22%) across all dataset for the convolution network with enlarged (dilated) receptive field. The largest change observed is the influence of the augmentation provided by the LSTM layers. In Table 7, this improvement in score (FD001- 4: 99.21%, 97.93%, 99.91%, 97.72% respectively) and RMSE (FD001- 4: 33.65%, 20.5%, 57.10%, 13.19% respectively) shows that the LSTM argumentation contributes the largest single improvement to the model performance. To further evaluate the contribution of the cross-validation approach implemented, the table also shows significant improvement in the score metric (52.57%, 83.68%, 78.77%, 95.94%) and RMSE metric (9.14%, 67.01%, 43%, 54.31%) over dataset FD001-4 respectively.

In summary, the best model performance is obtained with six (6) convolution layer model. Moreover, there is a clear trend in performance improvement when the convolution network is dilated, and the model is augmented with the LSTM layers. Also, a significant performance improvement is attributed to the cross-validated model training approach utilized in this work. Cross-validated models have been found to generalize better on out-of-sample datasets due to the robust training dataset distribution. In the implementation of the proposed method, the non-cross validated model performs worse than the proposed cross-validated model.

4.3. Comparison with WaveNet

Since the current work is inspired by the Wavenet architecture, it is pertinent to compare the output of the two models. In this section, the WaveNet model is extensively evaluated and compared with the proposed CaConvNet model. Also, the computational burden of the two models is presented.

The WaveNet model compared in this section consists of a stack of eight one-dimensional convolution layers, with similar kernel and filter as in the CaConvNet model. The number of trainable parameters for CaConvNet and WaveNet is 463,333 and 199,921 respectively. A similar loss function, cross-validation approach, and callback functions are implemented for both models. Table 8 and Fig. 11 shows the performance comparison of the two models. To measure the confidence in the models' statistical conclusions and the computation burden, Table 8 also shows the mean and standard deviation of the RMSE and Score metrics, and the time it takes to train both models using the C-MAPSS dataset on intel core i-7 workstation with RTX 2060s GPU.

It can be seen from Table 8 that WaveNet have less computation burden generally. However, the proposed CaConvNet performs better than the WaveNet on datasets FD001, FD003 and FD004. For the FD002 dataset, the WaveNet performs better both on the score (23.69%) and RMSE (7.62%) metric. However, the score and RMSE standard deviation for the WaveNet on FD002 is higher than that of the CaconvNet. The standard deviation shows high variability in the WaveNet prediction on FD002. This also reflects in the model prediction on the test set shown in Fig. 11. In Fig. 11(a & b), a small deviation is observed in the prediction between the proposed CaConvNet and the Wavenet. In Fig. 11(c, d), a larger deviation is seen in the WaveNet RUL prediction vs the real RUL for unit # 11 and #31 respectively. This is consistent with the high variability in the model prediction result in Table 8. WaveNet model prediction for units #31 and #96 of FD003 is close to the CaConVNet prediction, although CaConvNet prediction is smoother than WaveNet's. A similar trend is observed in the WaveNet vs CaConvNet prediction on unit #1 and #37 of FD004, shown in Fig. 11(g, h). The better CaConvNet prediction shown by the metric (score and RMSE) in Table 8 also reflects in the smoother trend in Fig. 11.

Table 8
Performance comparison of the proposed CaConvNet with the WaveNet model.

		RMSE			Score	
	Data	Value	Std. Dev	Training time (s)	Value	Std. Dev
WaveNet	FD001	13.84	0.416	505.67	106.27	7.344
	FD002	**14.05**	4.11	1734	**183.11**	168.15
	FD003	11.07	3.49	615.46	69.46	3.06
	FD004	27.55	1.09	1983.35	2257.288	34.15
Proposed CaConvNet	FD001	**11.83**	0.59	1058.10	**84.83**	0.416
	FD002	15.12	0.740	2098.80	226.33	4.11
	FD003	**9.63**	2.22	1395.46	**63.64**	3.49
	FD004	**22.31**	1.51	6139.13	**1179.63**	1.09

Table 9
Performance comparison of the proposed CaConvNet with conventional deep learning models.

Model	FD001		FD002		FD003		FD004	
	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE
Vanilla CNN	8765.12	20.18	6303.73	21.06	46381.49	23.82	17243.83	25.60
DLSTM	118.48	13.52	288.41	18.401	177.04	13.95	1253.91	23.22
CNN+LSTM	115.57	13.91	357.17	19.926	173.39	11.68	4871.53	23.27
DCAE	18259.22	20.78	28425.41	25.810	4661.47	12.55	77895.13	29.72
CaConvNet	**84.83**	**11.83**	**226.33**	**15.12**	**55.52**	**9.24**	**1179.63**	**22.31**

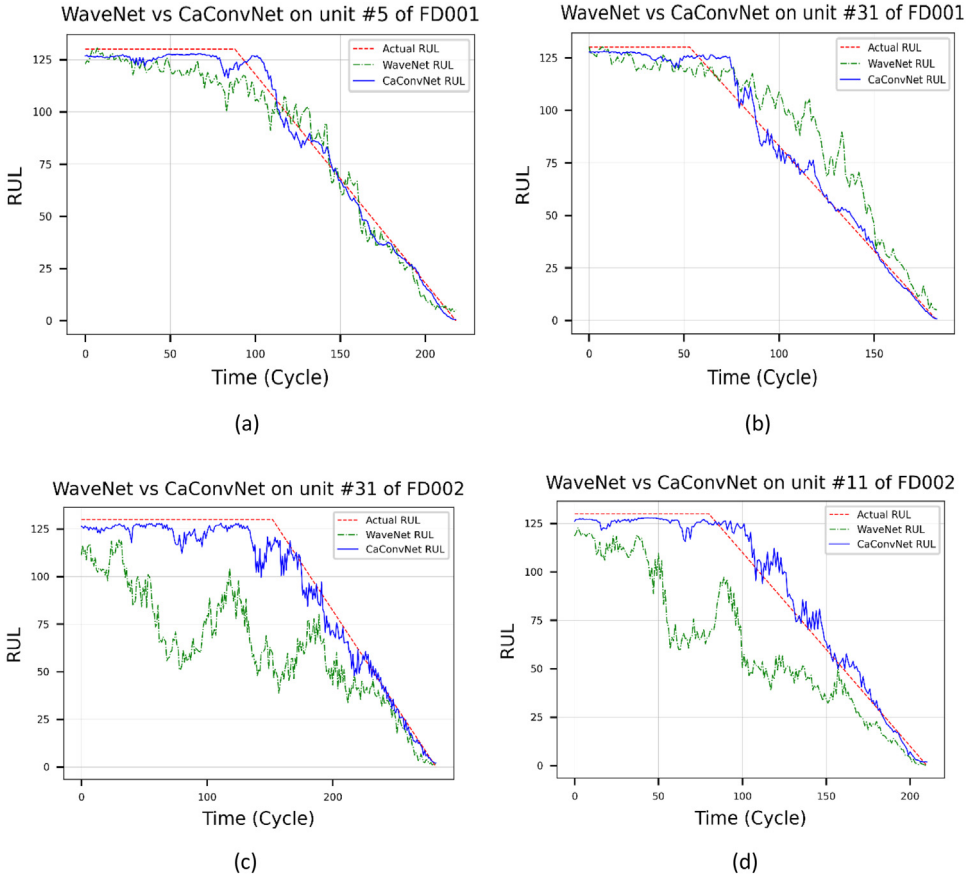

Fig. 11. (a) & (b) are the WaveNet vs CaConvNet RUL prediction for engine unit no 5 and 31 in FD001, (c)&(d) are the WaveNet vs CaConvNet RUL prediction for engine unit no 11 and 31 in FD002; (e) & (f) are the WaveNet vs CaConvNet RUL prediction for engine units 34 & 96 in FD003; and (g) &(h) are the predictions for engine units 1 and 37 in FD004 respectively.

4.4. Comparison with other conventional deep learning models

Table 9 presents the performance of conventional models trained using similar metrics used for CaConvNet. The evaluated models are the basic convolution neural network (vanilla CNN), deep long-short term memory network (DLSTM), convolution neural network with long short-term memory (CNN+LSTM), and the deep convolution autoencoder (DCAE). All models are cross-validated, and their training hyperparameters are similar to the ones used for CaConvNet. It is observed from Table 9 that the DLSTM and the CNN+LSTM model performances are better than other conventional models and are closer to the state-of-the-art performance of the proposed model. It is also observed that the DCAE model performs worse across all evaluated datasets.

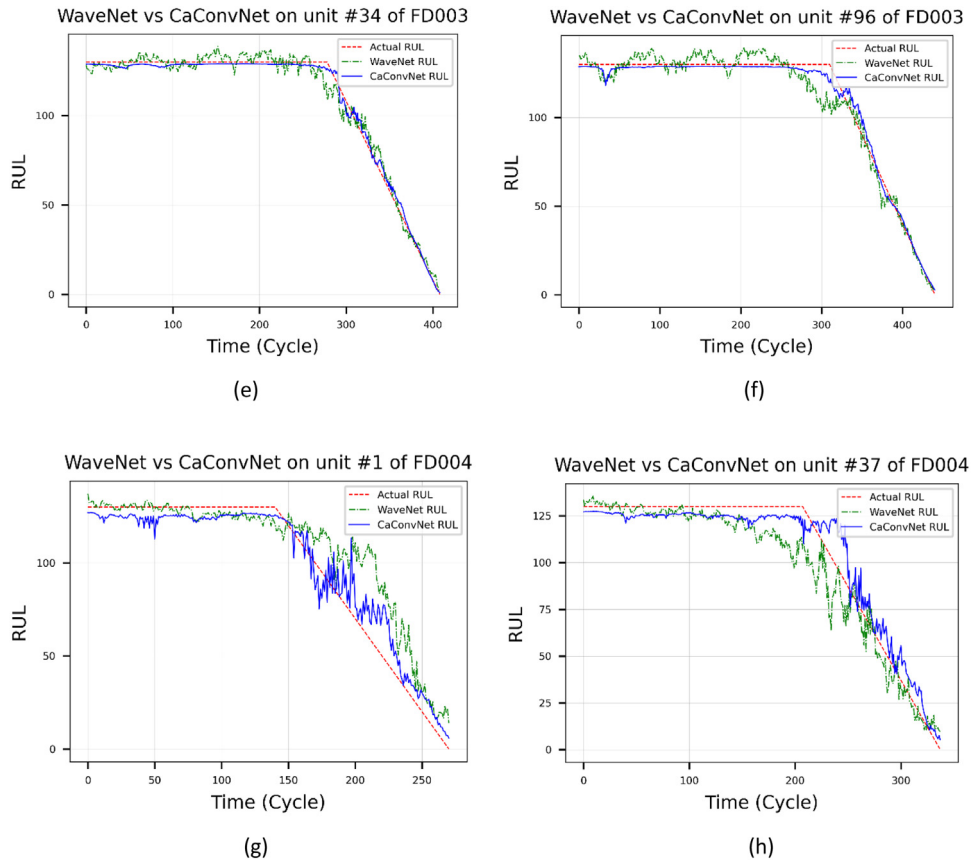

Fig. 11. (*continued*).

Moreover, the prediction comparison results between CaConvNet and the conventional models across selected test engines are presented in Fig. 12(a)–(h). For easy analysis, the figures contain the predictions for the three best models. It is seen from Fig. 12 that the CNN+LSTM and DLSTM models have closer predictions to the proposed model when the test engines are closer to their end of life. However, the DLSTM and CNN+LSTM models are computationally expensive, as the proposed model takes 48.6% less time to train. Also, the performance improvement of the proposed model over the conventional models, and over the conventional training approach is consistently the best across all evaluation datasets. To further quantify the performance improvement and novelty presented in this work, the predictive output of CaConvNet is compared with state-of-the-art models in recent literature.

4.5. Comparison with benchmark models

The task of selecting the best model for a specific task from series of candidate models is non-trivial. Hence, for a prognostic task involving run-to-failure datasets, the convention is to compare candidate models based on their prediction accuracy and repeatability. In this section, seven different state-of-the-art approaches in recent literature are compared with the proposed method. The compared models have been reported to perform better on the C-MAPSS dataset. The comparison is to further show the performance improvements of the proposed CaConvNet. Specifically, the following methods and models are compared:

a. Deep Convolution Neural Network (DCNN): This implementation uses four stacked convolution layers with dropout layers to reduce overfitting. All the layers use *tanh* activation function and a fully connected layer is used for the required regression output.
b. Bi-directional Long-short term memory (BiLSTM): the implementation uses a bidirectional network with the LSTM cell for sequence learning, reported to have the capacity to exploit long-range information in both input directions. Two bidirectional layers with *tanh* activation are used with two fully connected layers with Relu activation to estimate the RUL.
c. Hybrid deep neural network (HDNN): The hybrid deep neural network used here are stacks of three convolution neural network and long-short term memory layers. A series of fully connected layers (fusion path) is used to aggregate the RUL. The convolution network uses max-pooling and dropout layers.
d. Deep convolution generative adversarial network (DCGAN): This approach presents the generative adversarial network developed with auto-encoders. A multi-stage LSTM and fully connected layers are used to predict the RUL. A four-layer convolution neural network is used as the encoder input, before connecting the RUL-estimating layers.
e. RBM + LSTM: This implementation uses stacks of restricted Boltzmann machine with linear Gaussian units at the pre-training stage. Two-layer long short-term memory is used to learn long-term dependencies in sequential data, and a fully connected layer is used to aggregate the output. The ReLu and Sigmoid activation functions are used as the input–output activation.
f. Bi-directional long-short term memory with autoencoder (BiLSTM +ED): An online–offline training approach applied to bidirectional LSTM with the encoder–decoder model is used to predict remaining useful life. The obtained health index is compared with the online health index using the similarity curve matching to determine the final RUL.

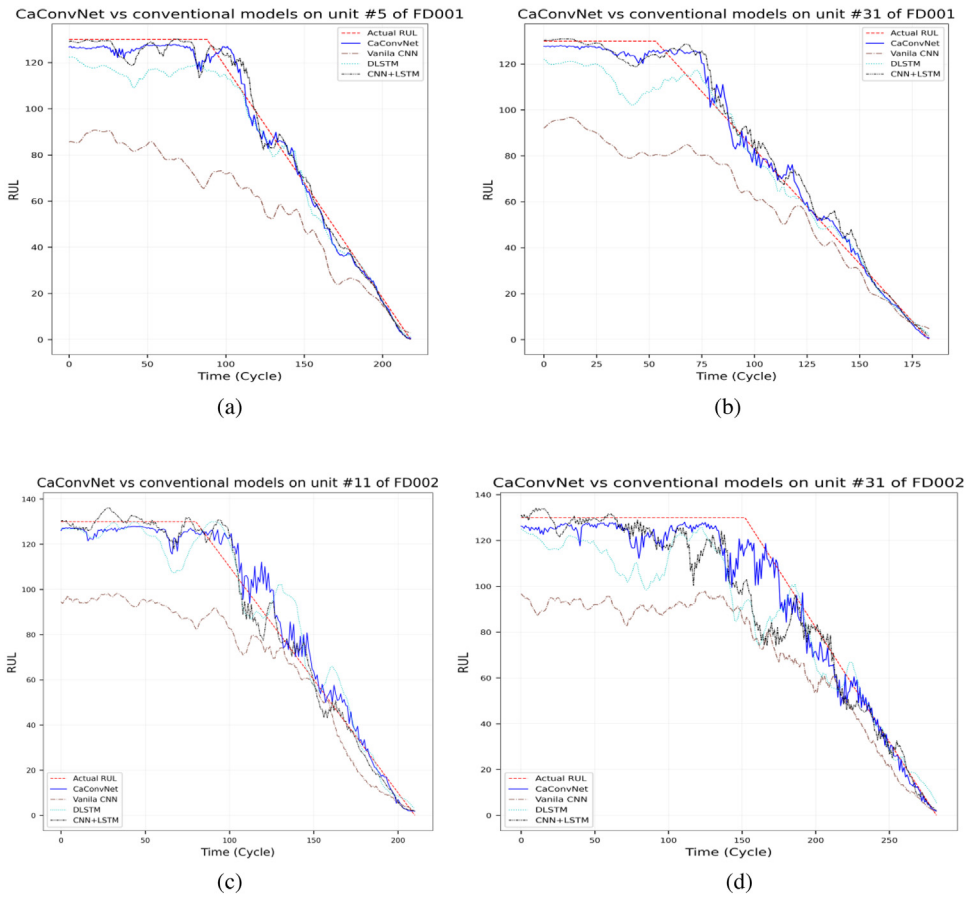

Fig. 12. (a) & (b) are the CaConvNet vs conventional models RUL prediction for engine unit no 5 and 31 in FD001, (c)&(d) are the CaConvNet vs conventional models RUL prediction for engine unit no 11 and 31 in FD002; (e) & (f) are the CaConvNet vs conventional models RUL prediction for engine units 34 & 96 in FD003; and (g) &(h) are the predictions for engine units 1 and 37 in FD004 respectively.

g. Rao-Blackwellized Particle Filter RBPF: Similarity-based RBPF method is a stochastic model that employs similar run-to-failure profile data as references to estimate the distribution of the degradation dataset and predict the RUL directly. The approach utilizes special techniques to exclude dissimilar run-to-failure dataset.

Tables 10 and 11 summarize the comparison result between CaConvNet and other models trained on CMAPSS dataset. Except for the similarity-based RBPF technique, all compared methods utilized the deep learning model and piece-wise RUL as the target. As shown in Table 10, it is observed that the proposed model has a significantly improved score value on all datasets compared with all analyzed state of the art models. For dataset FD001, FD002, FD003 and FD004, the proposed model has 51.23%, 92.41%, 74.65%, and 22.77% reduction in the score value compared with the best state-of-the-art model. A similar trend is noticed for the RMSE metric of the proposed model and the best state-of-the-art model. Apart from the DCGAN [48] and HDNN [35], which has a slight reduction in the RMSE value on FD001 and FD002 datasets, the proposed model achieved 0.79% and 16.12% reduction in the RMSE value on FD002 and FD003 datasets, compared with the best performing model. It is also observed that most implementation did not apply cutting edge model training techniques such as saving the best model while training or implementing *earlystopping callbacks* to reduce the risk of overfitting. Although rigorous testing of the proposed model on real in-service components is necessary, the presented result shows promising model performance in production.

5. Conclusion

This paper introduces the causal, augmented convolution network (CaConvNet), a new predictive model suitable for long sequence time series prediction. The proposed model combines the speed and lightness of dilated causal convolution networks with the order-sensitivity and temporal memory of the LSTM to predict the dynamics inherent in multidimensional, multi-timestep time series dataset. Also, auto-tuned implementation of the proposed model is presented, to enable flexibility and automate hyperparameter selection. This optimized implementation saves a lot of resources and reduces the uncertainty introduced by manually selecting the optimum parameters. Operating directly on a medium-size dataset that contains long-sequence signatures in run-to-failure turbofan engines, the model gives an improved prediction. Also, the superiority of the proposed model is verified through comparison with state-of-the-art models. This work contributes to knowledge by addressing the following

1. The work shows that the CaConvNet can learn and predict the dynamic inherent in noisy time series signals common in predictive maintenance tasks.

2. To deal with long-range temporal dependence and increase the network receptive field, a new architecture is developed based on dilated, temporal memory, and causal convolution layers.

3. The work also presents heuristic modifications and optimization techniques necessary to obtain an effective deep learning model suitable for remaining useful life prediction and other predictive maintenance tasks

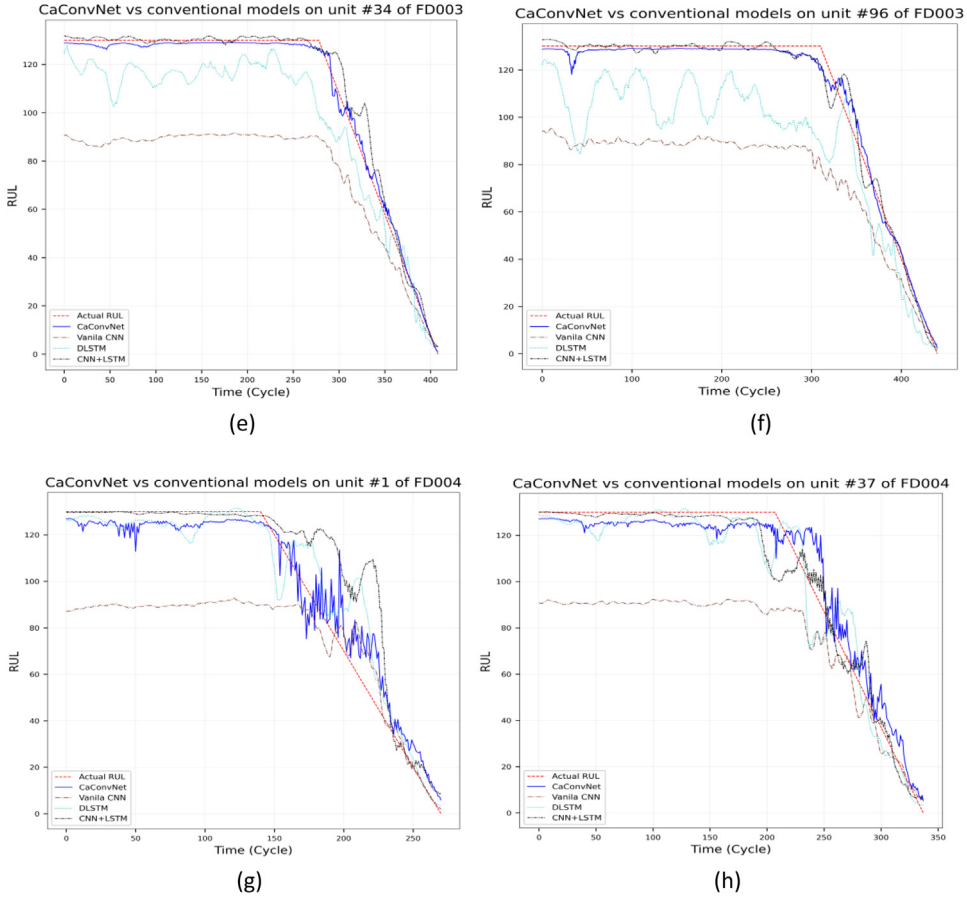

Fig. 12. (*continued*).

Table 10
Score comparison with recent benchmark models.

Source	Method	FD001	FD002	FD003	FD004
RESS [32]	DCNN	273.70	10412	284.10	12466
IEEE [34]	BiLSTM	295	4130	317	5430
IEEE [35]	HDNN	245	1282.42	1527.42	*1527.42**
Comput. Intell. Neurosci. [46]	*DCGAN*	*174**	2982	273	3874
RESS [29]	RBM+LSTM	231	3366	*251**	2840
MSSP [47]	BiLSTM+ED	273	3099	574	3202
Appl. Soft Comput [48]	RBPF	383.39	*1226.97**	375.29	2071.51
Current work	CaConvNet	**84.83****	**226.33****	**63.64****	**1179.63****

*Best score results in the literature.
**Current best score.

Table 11
RMSE comparison with recent benchmark models.

Source	Method	FD001	FD002	FD003	FD004
RESS [32]	DCNN	12.61	22.36	12.64	23.31
IEEE [34]	BiLSTM	13.65	23.18	13.74	24.86
IEEE [35]	HDNN	13.02	*15.24**	12.22	**18.16****
Comp. Intel. Neurosci [46]	*DCGAN*	**10.71****	19.49	*11.48**	19.71
RESS [29]	RBM+LSTM	12.56	22.73	12.10	22.66
MSSP [47]	BiLSTM+ED	14.47	22.07	17.48	23.49
Appl. Soft Comput [48]	RBPF	15.94	17.15	16.17	20.72
Current work	CaConvNet	11.83	**15.12****	**9.63****	22.31

*Best RMSE results in the literature.
**Current best RMSE.

4. The same architecture shows impressive results across different signatures and patterns presented in the four data subsets of the CMAPSS turbofan engine dataset. Moreover, evaluation results show a superior performance over other state-of-the-art models.

The CaConvNet architecture presented in this work provides a generic and flexible framework for tackling many tasks that use a time series dataset with long sequences, where temporal position and ordering is consequential. As a limitation of this work, the

most common contributor to component failure may not be reducible to the natural degradation of the component itself. Hence, apart from the degradation signals used to train this model, other features may be critical for optimized predictive maintenance. Moreover, deep learning model performance in production generally degrades with time as a result of concept drift, among other reasons. Therefore, to compensate for drifts, an effective pipeline to enable real-time retraining and fast redeployment of the model is required. As a future research focus, more rigorous testing of the proposed model on an in-service heavy industrial component dataset will be done to ensure consistent model performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62073288, 12075212, 61590921), National Key R&D Program of China (Grant No. 2018YFB2004200) and the Fundamental Research Funds for the Central Universities, China (Zhejiang University NGICS Platform) and their supports are thereby acknowledged.

References

[1] Aliabadi MM, Emami H, Dong M, Huang Y. Attention-based recurrent neural network for multistep-ahead prediction of process performance. Comput Chem Eng 2020. http://dx.doi.org/10.1016/j.compchemeng.2020.106931.
[2] Liu Y-k, Zhou W, Ayodeji A, Zhou X-q, Peng M-j, Chao N. A multilayer approach to DN 50 electric valve fault diagnosis using shallow-deep intelligent models. Nucl Eng Technol 2020. http://dx.doi.org/10.1016/j.net.2020.07.001.
[3] Wang H, Peng M-j, Ayodeji A, Xia H, Wang X-k, Li Z-k. Advanced fault diagnosis method for nuclear power plant based on convolutional gated recurrent network and enhanced particle swarm optimization. Ann Nucl Energy 2020;151:107934.
[4] Feng K, et al. Vibration-based updating of wear prediction for spur gears. Wear 2019;426:1410–5.
[5] Feng K, Smith WA, Borghesani P, Randall RB, Peng Z. Use of cyclostationary properties of vibration signals to identify gear wear mechanisms and track wear evolution. Mech Syst Signal Process 2021;150:107258.
[6] Zhang W, Guo W, Liu X, Liu Y, Zhou J, Li B, Lu Q, Yang S. LSTM-Based analysis of industrial IoT equipment. IEEE Access 2018;6:23551–60.
[7] Liu X, He S, Gu Y, Xu Z, Zhang Z, Wang W, Liu P. A robust cutting pattern recognition method for shearer based on least square support vector machine equipped with chaos modified particle swarm optimization and online correcting strategy. ISA Trans 2020;99:199–209.
[8] Cheng Y, Wang Z, Zhang W, Huang G. Particle swarm optimization algorithm to solve the deconvolution problem for rolling element bearing fault diagnosis. ISA Trans 2019;90:244–67.
[9] Ayodeji A, Liu Y-k. Support vector ensemble for incipient fault diagnosis in nuclear plant components. Nucl Eng Technol 2018;50(8):1306–13.
[10] He S, Xiao L, Wang Y, Liu X, Yang C, Lu J, Gui W, Sun Y. A novel fault diagnosis method based on optimal relevance vector machine. Neurocomputing 2017;267:651–63.
[11] He S, Liu X, Wang Y, Xu S, Lu J, Yang C, Zhou S, Sun Y, Gui W, Qin W. An effective fault diagnosis approach based on optimal weighted least squares support vector machine. Can J Chem Eng 2017;95(12):2357–66.
[12] Feng H-M, Wong C-C, Horng J-H, Lai L-Y. Evolutional RBFNs image model describing-based segmentation system designs. Neurocomputing 2018;272:374–85.
[13] Zhao J, Geng X, Zhou J, Sun Q, Xiao Y, Zhang Z, Fu Z. Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems. Knowl-Based Syst 2019;166:132–9.
[14] Ayodeji A, Liu Y-k, Xia H. Knowledge base operator support system for nuclear power plant fault diagnosis. Prog Nucl Energy 2018;105:42–50.
[15] Feng H-M, Chou H-C. Evolutional RBFNs prediction systems generation in the applications of financial time series data. Expert Syst Appl 2011;38(7):8285–92.
[16] Wang W, Zhang M, Liu X. Improved fruit fly optimization algorithm optimized wavelet neural network for statistical data modeling for industrial polypropylene melt index prediction. J Chemometr 2015;29(9):506–13.
[17] Xue T, Ding SX, Zhong M, Li L. A distribution independent data-driven design scheme of optimal dynamic fault detection systems. J Process Control 2020;95:1–9.
[18] Liu X, Gu Y, He S, Xu Z, Zhang Z. A robust reliability prediction method using weighted least square support vector machine equipped with chaos modified particle swarm optimization and online correcting strategy. Appl Soft Comput 2019;85:105873.
[19] Li L, Ding SX. Optimal detection schemes for multiplicative faults in uncertain systems with application to rolling mill processes. IEEE Trans Control Syst Technol 2019;28(6):2432–44.
[20] Ayodeji A, Liu Y-k. SVR Optimization with soft computing algorithms for incipient SGTR diagnosis. Ann Nucl Energy 2018;121:89–100.
[21] Zhou J, Yang Y, Ding SX, Zi Y, Wei M. A fault detection and health monitoring scheme for ship propulsion systems using SVM technique. Ieee Access 2018;6:16207–15.
[22] Ayodeji A, Liu Y-k, Zhou W, Zhou X-q. Acoustic signal-based leak size estimation for electric valves using deep belief network. In: 2019 IEEE 5th International Conference on Computer and Communications (ICCC). IEEE; 2019, p. 948–54.
[23] Pour FK, Theilliol D, Puig V, Cembrano G. Health-aware control design based on remaining useful life estimation for autonomous racing vehicle. ISA Trans 2020. http://dx.doi.org/10.1016/j.isatra.2020.03.032.
[24] Wang Y, Cang S, Yu H. Mutual information inspired feature selection using kernel canonical correlation analysis. Exp Syst Appl: X 2019;4:1000–14.
[25] Xiang S, Qin Y, Zhu C, Wang Y, Chen H. Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction. Eng Appl Artif Intell 2020;91:1035–87.
[26] Ahmad W, Khan SA, Islam MM, Kim J-M. A reliable technique for remaining useful life estimation of rolling element bearings using dynamic regression models. Reliab Eng Syst Saf 2019;184:67–76.
[27] Wen P, Zhao S, Chen S, Li Y. A generalized remaining useful life prediction method for complex systems based on composite health indicator. Reliab Eng Syst Saf 2021;205:107241.
[28] Chen J, Jing H, Chang Y, Liu Q. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. Reliab Eng Syst Saf 2019;185:372–82.
[29] Ellefsen AL, Bjørlykhaug E, Æsøy V, Ushakov S, Zhang H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab Eng Syst Saf 2019;183:240–51.
[30] Shi Z, Chehade A. A dual-LSTM framework combining change point detection and remaining useful life prediction. Reliab Eng Syst Saf 2020. http://dx.doi.org/10.1016/j.ress.2020.107257.
[31] Zhang W, Zhang Y, Zhai J, Zhao D, Xu L, Zhou J, Li Z, Yang S. Multi-source data fusion using deep learning for smart refrigerators. Comput Ind 2018;95:15–21.
[32] 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.
[33] Berghout T, Mouss L-H, Kadri O, Saïdi L, Benbouzid M. Aircraft engines remaining useful life prediction with an adaptive denoising online sequential extreme learning machine. Eng Appl Artif Intell 2020;96:103936.
[34] Wang J, Wen G, Yang S, Liu Y. Remaining useful life estimation in prognostics using deep bidirectional lstm neural network. In: 2018 IEEE prognostics and system health management conference (PHM-Chongqing); 2018.
[35] Al-Dulaimi A, Zabihi S, Asif A, Mohammadi A. Hybrid deep neural network model for remaining useful life estimation. In: ICASSP 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP); 2019.
[36] Wu J, Hu K, Cheng Y, Zhu H, Shao X, Wang Y. Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. ISA Trans 2020;97:241–50.
[37] Oord Avd, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. Wavenet: A generative model for raw audio. 2016, arXiv preprint arXiv:1609.03499.
[38] Borovykh A, Bohte S, Oosterlee CW. Dilated convolutional neural networks for time series forecasting. J Comput Finance 2019;22:73–101.
[39] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016.
[40] Zhao Z, Li T, Wu J, Sun C, Wang S, Yan R, Chen X. Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study. ISA Trans 2020;107:224–55.
[41] Li T, Zhao Z, Sun C, Yan R, Chen X. Multi-receptive field graph convolutional networks for machine fault diagnosis. IEEE Trans Ind Electron 2020.
[42] Araujo A, Norris W, Sim J. Computing receptive fields of convolutional neural networks. Distill 2019;4(11):21.

[43] Wang H, Peng M-j, Miao Z, Liu Y-k, Ayodeji A, Hao C. Remaining useful life prediction techniques for electric valves based on convolution auto encoder and long short term memory. ISA Trans 2021;108:333–42.
[44] Saxena A, Goebel K. Turbofan Engine Degradation Simulation Data Set. NASA Ames Prognostics Data Repository. Moffett Field.: NASA Ames Research Center; 2008.
[45] Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A. Hyperband: A novel bandit-based approach to hyperparameter optimization. J Mach Learn Res 2017;18(1):6765–816.
[46] Hou G, Xu S, Zhou N, Yang L, Fu Q. Remaining useful life estimation using deep convolutional generative adversarial networks based on an autoencoder scheme. Comput Intell Neurosci 2020. http://dx.doi.org/10.1155/2020/9601389.
[47] Yu W, Kim IY, Mechefske C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mech Syst Signal Process 2019;129:764–80.
[48] Cai H, Feng J, Li W, Hsu Y-M, Lee J. Similarity-based particle filter for remaining useful life prediction with enhanced performance. Appl Soft Comput 2020. http://dx.doi.org/10.1016/j.asoc.2020.106474.