WALAILAK JOURNAL
http://wjst.wu.ac.th Engineering and Physical Sciences

Remaining Useful Life Prediction Using Enhanced Convolutional Neural Network on Multivariate Time Series Sensor Data

Manassakan SANAYHA and Peerapon VATEEKUL[*]

Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330, Thailand

([*]Corresponding author's e-mail: peerapon.v@chula.ac.th)

Received: 30 April 2017, Revised: 20 December 2018, Accepted: 22 January 2019

Abstract

All machines in power plants need high reliability and to be continuous run at all times in the production process. The Remaining Useful Life (RUL) prediction of machines is an estimation for planning maintenance activities in advance to save the cost of corrective and preventive maintenance. Most existing models analyze sensor data separately. This univariate analysis never considers the relationship between sensors and time simultaneously. In this paper, we applied a Convolutional Neural Network (CNN), which considered both dimensions of and sensors; a multivariate time series analysis. Furthermore, we applied many techniques to enhance the framework of deep learning, including dropout, L^2 Regularization, and the Adaptive Gradient Descent (AdaGrad). For the experiment, we conducted our method and showed the performance in term of Root Mean Square Error (RMSE) on a standard benchmark and for real-case datasets.

Keywords: multivariate time series, deep learning, convolutional neural network, remaining useful life

Introduction

Any power plant requires higher efficiency, but unwanted downtime is often caused by machine fault problems. Thus, predictive maintenance is the most important activity for solving this problem, producing cost savings over corrective and preventive maintenance. The remaining useful life (RUL) prediction has become a major scientific challenge, a prognostic technique that aims to accurately estimate the RUL. All machine components have many points of sensor data. Existing algorithms, such as linear models, cannot capture the complex relationship between the sensor data and the RUL. Moreover, Multilayer Perceptron (MLP) cannot learn the salient features automatically because of its network structure. Therefore, estimating the RUL uses a Convolutional Neural Network (CNN) based regression approach.

Baraldi *et al.* [1] proposed an original method to extend Particle Filtering (PF) in the case of an analytical measurement model. The PF scheme was applied to a case study regarding the prediction of the RUL of a structure which was degrading, according to a stochastic fatigue crack growth model from the literature. Porotsky [2] presented the Cross-Entropy method for controlling parameter optimization, based on the Cross-Validation procedure. The solution was recognized as a winner in when used in competition. The results demonstrated the effectiveness of the approach for the RUL estimation for system parameters with non-trend ability behavior. Javed *et al.* [3] applied wavelet-extreme learning machine and subtractive-maximum entropy fuzzy clustering to predict the RUL of a machine using simultaneous predictions and discrete state estimation. The model objectively assigned dynamic failure threshold procedure to estimate the RUL.

Yang *et al.* [4] compared the 2 RUL prediction approaches, which were the Back Propagation-Artificial Neural Network (BP-ANN) and the Extreme Learning Machine (ELM), on the popular turbofan engine degradation dataset, and evaluated the performance to show that the BP-ANN outperformed ELM, but took more time to train the model. Ren *et al.* [5] combined time and frequency domain features, using a deep learning approach, to predict the RUL of multi-bearing, which could extract high-quality degradation patterns from vibration signals of the rolling bearings using a neural network-based model and rectified linear and sigmoid activation function. Root Mean Square Propagation (RMSprop) optimization was used for minimizing the loss function in their proposed model. The performance was evaluated on real datasets with other commonly used prediction methods to demonstrate the effectiveness of the model. Zhao *et al.* [6] proposed Adjacent Difference Neural Network (ADNN), which added the adjacent term in loss function to smooth the weight in the network for the RUL prediction model. Babu *et al.*[7] firstly proposed a novel CNN based regression approach. They applied it along the temporal dimension over the multi-channel sensor data through the deep architecture. The feature learning and the RUL estimation were mutually enhanced by the supervised feedback. The results were more efficient and accurate than several state-of-the-art algorithms.

In this paper, an RUL prediction model, using a data-driven approach called enhanced CNN, which is suitable for multivariate time series data from machines or equipment with continuous in timely order data and multi-sensors, is proposed. The structure of the prediction model consists of deep learning architecture with many optimization techniques, such as dropout, L^2 Regularization, and the AdaGrad. Moreover, this model has efficient feature selection and temporal at the same time point, which is significant for the RUL prediction task in order to maintain the machine for efficiency in power plants or manufacturing.

This paper is organized as follows. Section I is the introduction, and Section II presents background knowledge about CNN. Section III is the proposed method. Section IV describes the experimental data, composed of 4 datasets, and Section V presents the experiments and results. Finally, Section VI is the conclusion.

Background Knowledge

Convolutional neural network

CNN is a special kind of multi-layer neural networks, designed to recognize visual patterns directly from pixel images with minimal preprocessing. It can recognize patterns with extreme variability, and with robustness to distortions and simple geometric transformations. The structure of a CNN, shown in **Figure 1**, is a simple, well-known architecture called "LeNet", introduced by Lecu *et al.* in 1998 [8].

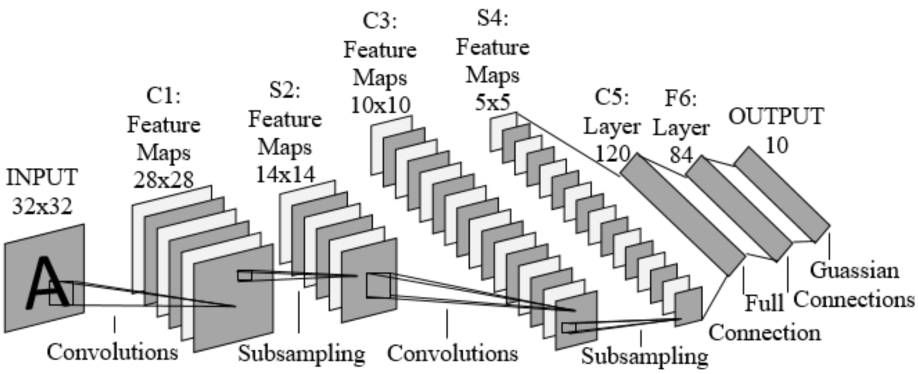

Figure 1 Structure of the well-known CNN, LeNet [8].

The structure of CNN consists of 4 important layers, as per the following.

1) Convolutional Layer: The first layer is connected after the input. The array of numbers, known as the "Kernel Filter", which may be weighted or use parameters in this layer, will compute a dot product of input and weight for the output for the next layer.

2) Activation Function: The non-linearity function performs a mathematical operation on a single number. There are several activation functions which provide outputs for various problem solving.

I) Rectified Linear Unit Function (ReLU): The result of this activation function is I^+ or zero, as in (1).

$$f(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases} \tag{1}$$

II) Exponential Linear Unit Function (ELU): This activation function is computed with an exponential function, as in (2).

$$f(\alpha, x) = \begin{cases} \alpha(e^x - 1)\ , if\ x < 0 \\ x\ , if\ x \geq\ 0 \end{cases} \tag{2}$$

3) Pooling Layer: This layer performs a down sampling operation along the dimensions of the input by computing a maximum or average value, called "Max Pooling" or "Average Pooling".

4) Fully Connected Layer: The final layer of the convoluted structure will compute the result, class, or score. All neurons in this layer are connected to the neurons in the previous layer.

Loss function

The Mean Square Error (MSE) is commonly used to compute the loss function, which measures the quality of a particular set of weights or parameters. The equation is indicated in (3).

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{3}$$

where J is the loss function, $\hat{y}_i$ is the predicted value of dataset I and y_i is the actual value of dataset i.

Optimization

The purpose of optimization is to find the weight that minimizes the loss function.

1) Stochastic Gradient Descent (SGD): The learning with SGD will adjust the weight using previous information, according to (4).

$$w_t = w_{t-1} - \alpha\frac{\partial_{I_t}}{\partial_w} \tag{4}$$

where w is the adjusting weight, α is the learning rate and $\frac{\partial_I}{\partial_w}$ is the gradient of loss function to w.

2) The Adaptive Gradient Descent (AdaGrad): This method will adjust the learning rate by itself. Then, all the previous gradients will be used to adjust the learning rate in (5) and (6).

$$g_t = \frac{\partial_I}{\partial_w} \tag{5}$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{\sum_{k=1}^{t} g_{i^2}}} g_t \tag{6}$$

where g_t is the gradient at the time t and w_t is the adjusting weight of time t.

Overfitting problem

This problem occurs when the model learns the detail and noise in the training data too well. Therefore, it impacts the performance of the model with new data, which means that the model cannot be applied to new data and be generalized.

1) L^2 Regularization: The most common form of regularization, by adding the sum of squared weight values to the loss function term while training, is shown in (7).

$$C = C_0 - \frac{\lambda}{2n}\sum_w w^2 \tag{7}$$

where C_0 is the initial loss function and λ is the regularization parameter.

2) Dropout: This technique prevents overfitting by making the co-adaptations on training data more complex. The dropout will randomly remove some of the units to create a different network, which is trained by backpropagation.

Proposed method

Our prediction procedure is composed of training data, preprocessing, prediction model, and performance evaluation, as shown in **Figure 2**.

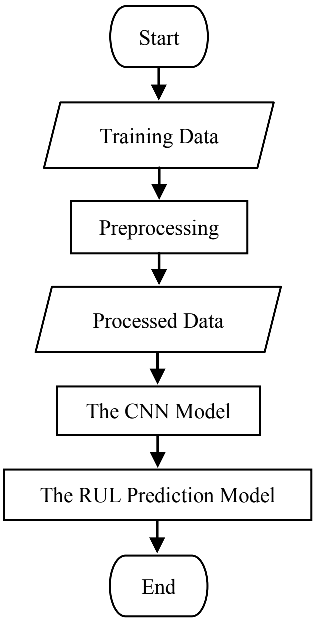

Figure 2 Process of the RUL prediction model.

Preprocessing

It is necessary to normalize the input data, because each variable has a very different range of value; for example, the *1st* sensor point is the main range, while the *8th* sensor point is the thousands range. To find a correlation, or multivariate analysis, in the next step, the normalization method, we apply the z-score according to (8), which represents the distance between the single values and the population mean in units of the standard deviation.

$$z = (x - \mu)/\sigma \tag{8}$$

where μ is the mean of the input x and σ is the standard deviation of the input x.

The remaining useful life prediction model

From a study of relevant research, it can be seen that the neural network which is the best method to predict the RUL is CNN. The structure uses the exponential rectified function as an activation function in the feature map, max pooling, and adaptive gradient descent as a loss function. Moreover, we use L^2 regularization to prevent overfitting of the model. The structure of the prediction is shown in **Figure 3**. The input is a 2-dimension array in $S\times T$ when S is the number of sensor signals which depends on datasets and T is the number of time points; 15 time points are used. The convolutional, pooling, and fully connected layers are set in the structure according to the standard VGG-16 model. After the convolutional layers, we apply the following techniques; dropout, L^2 regularization, and the AdaGrad. From **Figure 3**, the prediction of the RUL values is calculated after the CNN model. Our approach focuses on introducing new techniques to reduce predictive errors, as follows:

- Restructuring the CNN in convolutional, pooling, and fully connected layers for the appropriation of multivariate and time series data.
- Model optimization by changing the activation function from the commonly used sigmoid function as an exponential linear unit function, which will help to cut the core down because, if the value is less than zero, the values from this function will be negative, according to the exponential trend.
- Implementing techniques to prevent referencing with learning information or dropout is used to randomly cut out unnecessary links. The L^2 regularization reduces replication of the model after finding the feature extraction. These 2 overfitting prevention techniques are only done while training the model, and not in the testing process.
- Applying the AdaGrad to optimize the loss function by adjusting the weights using all the previous gradients, which provide a preferable learning rate. This will affect the loss function and the predicted output.

Model configuration

In this step, we explain how to configure and adjust our CNN model for the RUL prediction. The model training is an important step. The dimension of input is $S\times T$; the number of sensors S and 15 time points T. The setting of batch size is 10 and number batches per epoch are 800. Moreover, the numbers of training are 2,000, 2,500, and 5,000 epochs per cycle to evaluate the number of rounds, which gives the least error. The execution time in each round of epoch depends on the size of input datasets and models.

For the testing, we evaluate the most efficient model by measuring the RMSE to indicate the difference between the actual and predicted values for performance evaluation of the model.

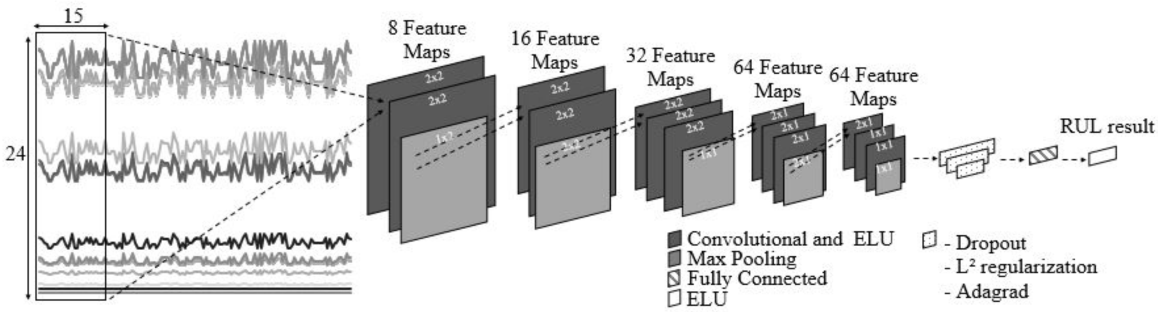

Figure 3 Structure of proposed CNN for remaining useful life prediction.

Experimental data

In our experiment, there are 4 time series datasets on remaining useful life prediction, comprising of (*i*) 3 public datasets, which are reliable and well-known in the prognostic fields, and (*ii*) one real-cases from a power plant, for empirical prediction in real world situations.

Prognostic data challenge dataset

This dataset was used for a prognostics challenge competition at the International Conference on Prognostics and Health Management (PHM08) [9]. Data sets consist of multiple multivariate time series.

Input data was a Turbofan Engine Degradation Simulation, which was prepared by the Prognostics Center of Excellence, or PCoE, at the NASA Ames Research Center. It was made by Commercial Modular Aero-Propulsion System Simulation, or C-MPASS. The dataset was collected under the conditions of several sensor points, symptoms of equipment failure, and working environment. It certainly included disturbing information. **Figure 4** shows the components of the turbofan and the variable names and additional metric description of the turbofan series.

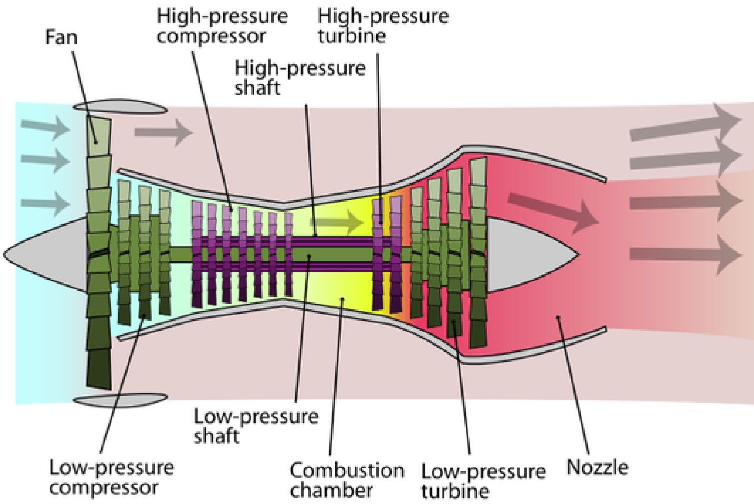

Figure 4 Components of turbofan [10].

Each engine starts with different initial conditions and manufacturing variations. The engine operates normally until it starts to degrade at some time point during the simulation. The degradation grows until it reaches a setting threshold, where it is not preferable to continue to operate the machine during the simulation.

For the information of the datasets, **Tables 1** and **2** show the variable names and their descriptions. There are 3 operational settings that affect engine performance, and 21 sensors. **Table 3** contains a number of observed turbofans in each sub-dataset.

PHM08 pronostic data challenge

This dataset is similar to *A*. The dataset was used for the prognostics challenge competition at the International Conference on Prognostics and Health Management (PHM08). This challenge is open for the researchers to develop and compare their model and performance against others with score function; the Mean Square Error (MSE).

Table 1 Name and description database of input data.

No.	Attribute	Description
1	unit no.	the engine number
2	Time	the operational cycle number
3	opt.set1	the 3 operating settings
4	opt.set2	
5	opt.set3	
6	sensor1	the 21 sensor values
7	sensor2	
8	sensor3	
...	...	
26	sensor21	

Table 2 Symbols and description of sensor signals for turbofan engine degradation dataset.

NO.	Symbol	Description	Units
1	T2	Total temperature at fan inlet	°*R*
2	T24	Total temperature at LPC outlet	°*R*
3	T30	Total temperature at HPC outlet	°*R*
4	T50	Total temperature at LPT outlet	°*R*
5	P2	Pressure at fan inlet	*psia*
6	P15	Total pressure in bypass duct	*psia*
7	P30	Total pressure at HPC outlet	*psia*
8	Nf	Physical fan speed	*rpm*
9	Nc	Physical core speed	*rpm*
10	Epr	Engine Pressure ratio	-
11	Ps30	Static pressure at HPC outlet	*psia*
12	Phi	Ratio of fuel flow to Ps30	*pps/psi*
13	NRf	Corrected fan speed	*rpm*
14	NRc	Corrected core speed	*rpm*
15	BPR	Bypass ratio	-
16	farB	Burner fuel-air ratio	-
17	htBleed	Bleed enthalpy	-
18	Nf_dmd	Demanded fan speed	*rpm*
19	PCNfR_dmd	Demanded corrected fan speed	*rpm*
20	W31	HPT coolant bleed	*lpm/s*
21	W32	LPT coolant bleed	*lbm/s*

°*R* refers to the Rankine temperature scale.
psia refers to Pounds per square inch absolute.
rpm refers to Revolutions per minute.
pps refers to Pulse per second.
psi refers to Pounds per square inch.
lbm/s refers to Pound mass per second.

Table 3 Numbers of input for each sub-dataset.

Dataset	C-MAPSS				PHM08
	FD001	FD002	FD003	FD004	
Training	100	260	100	249	218
Testing	100	259	100	248	218

Virkler dataset

The dataset is fatigue crack growth under homogeneous cyclic stress for statistical analysis [11]. The amplitude test dataset in aluminum alloy was carried out to investigate fatigue crack propagation with 69 replicate constants, which were crack lengths in mm. and the observed series. The crack lengths were observed from the number of cycles, while testing was as shown in **Figure 5**.

Circulating water pump (CWP)

In a steam turbine generator system, there is a heat exchange medium that supplies through a condenser, which is the main function of circulating water. The Circulating Water Pump (CWP), shown in **Figure 6**, will pump the circulating water to absorb heat from the system, then return it to the cooling tower.

Each CWP will be monitored by many sensor points, like 1 current, 3 temperatures and 4 vibration sensor points. Sensor data of CWP equipment are collected from a real-time data management tool called the PI system, which is used for data collection, historicizing, analyzing, delivering, and visualizing in plant maintenance tasks. Signals would have been collected at hourly of one year; one sensor point will have 8,760 (24×365) values. In this case, we collect 8 sensor points for a CWP. Finally, the sensor signal will have 61,320 input values. Sensor data of CWP equipment are collected from a real-time data management tool called the PI system, which is used for data collection, historicizing, analyzing, delivering, and visualizing in plant maintenance tasks. Signals are collected hourly for one year. One sensor has 8,760 (24×365) values. We collect 61,320 input values of 8 sensor points for a CWP for the experiment in our proposed method.

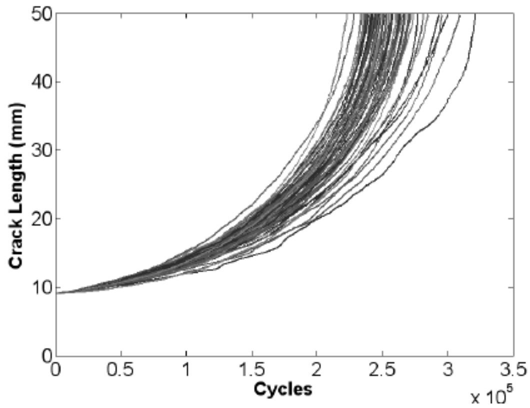

Figure 5 Crack length propagation samples under same loading conditions [12].

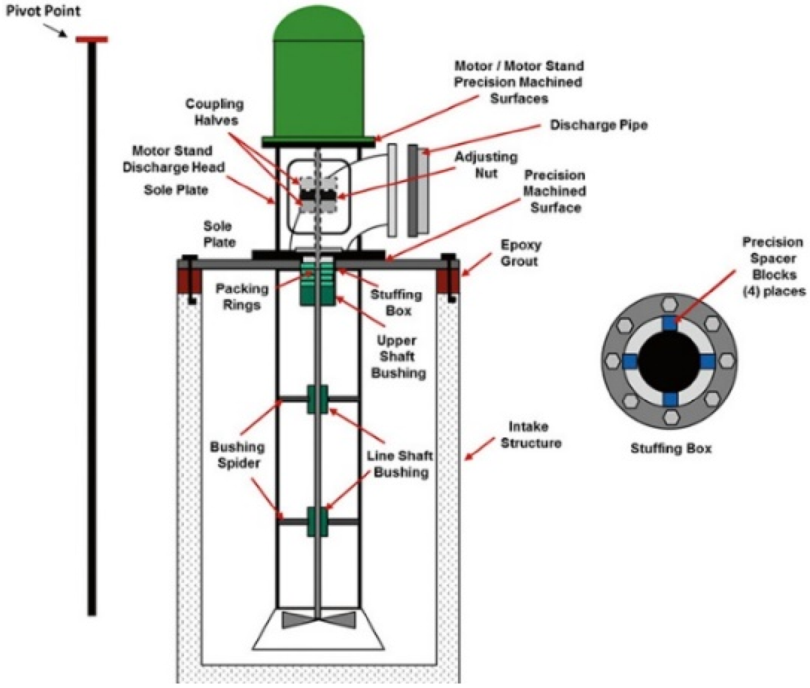

Figure 6 Structure of CWP components [13].

Experiments and results

In the experiment, the 4 datasets are used to predict the remaining useful life. We conduct 2 statistical methods for the univariate forecasting, the standard CNN, and our proposed method; multivariate forecasting, to predict the time series data. The univariate forecasting only predicts the RUL value as a single time series data, while multivariate forecasting predicts the RUL with the analysis of the correlation between input sensors.

Univariate forecasting

The experiment is performed by calculating the appropriate model equation using the well-known statistical Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing methods in forecasting the RUL value according to the consistency of the data and time. The analysis takes 30 days; 720 h or time points of input to predict the RUL values for one day; 24 time points. Then, we evaluate the performance by using the RMSE value, which calculates the difference between the predicted and actual values.

Convolutional neural network

The experiments are conducted in accordance with the presented method by the convolutional layers. This uses the sigmoid function to calculate the output data and the pooling twice and continue with the fully-connected layer. In the training phase, the optimized value is achieved with the AdaGrad in the evaluation of network parameters, and the back-propagation algorithm to reduce the loss function value. The error of prediction are summarized from all 3 methods, which are standard statistical time series forecasting: the ARIMA and Exponential Smoothing; the baseline methods: the standard CNN model, and the proposed methods: enhanced CNN. The enhanced techniques we apply in each CNN method in our experiments is described in **Table 4**. We apply the AdaGrad in every method, because the loss function cannot be optimized with the SGD.

The RMSE of prediction is calculated to measure the difference between the predicted and real values. **Table 5** represents the RMSE of each method, showing that our proposed method outperformed others, with the comparison of 3 public datasets and one real-case dataset, which makes the model more applicable.

Based on the experiment, we compare the efficiency of our proposed method with the standards; the statistical and CNN models. The predicted result with CNN gives less error than the statistic models, but our proposed enhanced CNN gives the least error, as shown in **Table 5**. In this work, we will focus on network restructuring and the application of methods such as dropout, L^2 regularization, and the AdaGrad to reduce the error of prediction.

Table 4 Techniques applied in each CNN method.

	Methods	Activation function	Dropout	L^2 regularization	Optimization
Standard CNN	Lenet-5 Lite	ReLU	×	×	AdaGrad
	Alexnet Lite	ReLU	×	×	AdaGrad
	VGG-16 Lite	ReLU	×	×	AdaGrad
Enhanced CNN	Enhanced Lenet-5	ELU	✓	✓	AdaGrad
	Enhanced Lenet-5-2	ELU	×	✓	AdaGrad
	Enhanced VGG-16	ELU	×	✓	AdaGrad

Table 5 RMSE of each dataset and methods An asterisk symbol (*) refers to the winning methods.

	Method	C-MAPSS1	C-MAPSS2	C-MAPSS3	C-MAPSS4	PHM08	Virkler	CWP
Univariate Forecasting	ARIMA	6.93	15.59	15.70	16.36	15.94	30.27	19.05
	Exponential Smoothing	7.16	16.56	16.66	17.37	16.87	31.94	20.15
Multivariate Forecasting	Lenet-5 Lite	19.52	29.41	20.93	31.16	31.01	6.68	48.44
	Alexnet Lite	8.89	10.77	11.76	15.74	17.51	5.59	45.24
	VGG-16 Lite	13.85	24.39	29.06	48.46	23.37	8.53	21.75
	Enhanced Lenet-5	18.24	28.72	18.83	30.34	32.19	18.91	49.04
	Enhanced Lenet-5-2	17.57	26.98	18.04	28.84	29.17	17.08	48.54
	Enhanced VGG-16	2.5*	6.41*	3.25E-03*	10.72*	6.26*	3.12*	19.02*

Conclusions

In this paper, we proposed the RUL prediction model, which helps future maintenance tasks to be more predictable and more accurate, using a machine learning method called CNN. The model is enhanced from the basic structure of deep learning and improves performance of prediction by using some techniques. The experiments were conducted with 3 standard benchmarks and one real-case dataset. Then, we compared the predicted result of each dataset with 8 prediction methods and evaluated the performance of a model using the standard evaluation RMSE, which compares the differences between the predicted and the real RUL values. Our proposed method outperformed others, with small prediction error. Therefore, we can have precise estimation.

From the results, we can apply our proposed model embedded in the real-world system to help as a pre-warning for equipment breakdown or malfunction which may occur in the future. Therefore, maintenance engineers can prepare resources to inspect or check the equipment, which can reduce the cost of maintenance.

References

[1] P Baraldi, M Compare, S Sauco and E Zio. Ensemble neural network-based particle filtering for prognostics. *Mech. Syst. Signal Process.* 2013; **41**, 288-300.
[2] S Porotsky. Remaining useful life estimation for systems with non-trendability behavior. *In*: Proceedings of the 2012 IEEE Conference on Prognostics and Health Management. 2012, p. 1-6.
[3] K Javed, R Gouriveau and N Zerhouni. A new multivariate approach for prognostics based on extreme learning machine and fuzzy clustering. *IEEE Trans. Cybernet.* 2015; **45**, 2626-39.
[4] Z Yang, P Baraldi and E Zio. A comparison between extreme learning machine and artificial neural network for remaining useful life prediction. *In*: Proceedings of the 2016 Prognostics and System Health Management Conference. 2016, p. 1-7.
[5] L Ren, J Cui, Y Sun and X Cheng. Multi-bearing remaining useful life collaborative prediction: A deep learning approach. *J. Manuf. Syst.* 2017; **43**, 248-56.
[6] Z Zhao, L Bin, X Wang and W Lu. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliab. Eng. Syst. Saf.* 2017; **164**, 74-83.
[7] GS Babu, P Zhao and XL Li. Deep convolutional neural network based regression approach for estimation of remaining useful life. *In*: Proceedings of the 21st International Conference Database Systems for Advanced Applications. 2016, p. 214-28.
[8] NY Lecu, ULB To, Y Bengio and RP Haffne. Gradient-based learning applied to document recognition. *Proc. IEEE* 1998; **86**, 2278-324.
[9] A Saxena, K Goebel, D Simon and N Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. *In*: Proceedings of the International Conference on Prognostics and Health Management. 2008, p. 1-9.
[10] Predictive Modeling Of Turbofan Degradation (Part 1). Data Scientist In A Box. Available at: https://datascientistinabox.com/2015/11/30/predictive-modeling-of-turbofan-degradation-part-1, , accessed June 2017.
[11] O Ditlevsen and R Olesen. Statistical analysis of the virkler data on fatigue crack growth. *Eng. Fract. Mech.* 1986; **25**, 177-95.
[12] OF Eker, F Camci and IK Jennions. Major challenges in prognostics: Study on benchmarking prognostics datasets. *In*: Proceedings of the 1st European Conference of the Prognostics and Health Management Society. 2012.
[13] Pumps & Systems. Steps to Successful Installation of Vertical Circulating Water Pumps, Available at: https://www.pumpsandsystems.com/pumps/november-2015-steps-successful-installation-vertical-circulating-water-pumps, accessed January 2019.