# Inception Based Deep Convolutional Neural Network for Remaining Useful Life Estimation of Turbofan Engines

Nathaniel DeVol[1], Christopher Saldana[2], and Katherine Fu[3]

[1,2] *George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA*
*ndevol3@gatech.edu*
*christopher.saldana@me.gatech.edu*

[3] *Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI, 53706, USA*
*kate.fu@wisc.edu*

**ABSTRACT**

Accurate estimation of the remaining useful life (RUL) is a key component of condition based maintenance (CBM) and prognosis and health management (PHM). Data-based models for the estimation of RUL are of particular interest because expert knowledge of systems is not always available and physical modeling is often not feasible. In this paper, a deep convolutional neural network (CNN) architecture is investigated for its ability to estimate the RUL of turbofan engines. The input to the model is a window of time series data collected from the engine under test. Inputting raw sensor data allows features to be learned instead of manually determined. To incorporate the ability to detect features of differing lengths, inception modules are used in the neural network architecture. The model is trained and tested using the new Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) data set and high prognosis accuracy was achieved. The developed model was used in the 2021 PHM Society Data Challenge and received second place, further validating its ability to accurately estimate RUL.

## 1. INTRODUCTION

Remaining useful life (RUL) is the remaining time for which an asset will be productive (Si, Wang, Hu, & Zhou, 2011), and the RUL is a key component of condition-based maintenance (CBM) and prognosis and health management (PHM) (Jardine, Lin, & Banjevic, 2006). Knowing the RUL of an asset allows for the optimal scheduling of maintenance and ordering of spare parts, which can reduce asset downtime and increase profitability. RUL estimation also has safety and environmental implications by preventing failures that could put users in danger and extending the life of an asset, thereby reducing the need for new equipment (Si et al., 2011).

While the RUL of an asset is affected by many variables and cannot be exactly known, there are several modeling techniques for estimating the RUL. The three primary approaches for estimating the RUL of an asset are: physics-based, data-based, and hybrid (Sikorska, Hodkiewicz, & Ma, 2011). Physics-based methods create a model of the system using an in-depth understanding of the underlying processes. While physics-based models have been shown to be successful (Bolander, Qiu, Eklund, Hindle, & Rosenfeld, 2009), they are often prohibitive due to the time and system understanding required to create them. Additionally, physics-based models tend to be specific to a failure mode, and failure modes must be well understood *a priori* (Sikorska et al., 2011). Data-based methods, including statistical and artificial intelligence (AI) methods, do not require knowledge of the underlying systems, but instead rely on the availability of a data set that captures the performance of the system. Finally, hybrid models merge these two approaches to reduce the underlying knowledge and data requirements of the two individual methods (Sikorska et al., 2011). While hybrid approaches work in some applications (Kong et al., 2020), they increase the modeling complexity by requiring the two models to be developed and merged.

In this work, a modeling methodology for estimating the RUL of turbofan engines is investigated. This was completed as part of the 2021 PHM Society Data Challenge, which used the new Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) data set (Chao, Kulkarni, Goebel, & Fink, 2021). This data set contains a large amount of run-to-failure data, so this paper takes a data-based approach. Past research on the original CMAPSS data set demonstrated the applicability of convolutional neural networks (CNNs) (Li, Ding, & Sun, 2018; Yang, Zhao, Jiang, Sun, & Mei, 2019), long short-term memory (LSTM) (da Costa, Akeay, Zhang,

& Kaymak, 2019; Zheng, Ristovski, Farahat, & Gupta, 2017; Wu et al., 2020), and hybrid methods, which merge CNN and LSTM (Zhao, Huang, Li, & Iqbal, 2020) for RUL estimation. The use of convolutions allows for a reduction of the number of model parameters and aids in the extraction of features (Albawi, Mohammed, & Al-Zawi, 2017). Recurrent networks, such as the LSTM, incorporate historical data so that changes over time can be realized (Hochreiter & Schmidhuber, 1997). The use of recurrent networks, however, requires the previous timesteps of data to be available for an estimation to be made. The previous data may not always be available, so in this work, CNNs are investigated for their ability to estimate the RUL of turbofan engines.

The rest of the paper is organized as follows. In section 2, the problem is introduced and the data set is described. In section 3, the methodology is detailed, including the data preprocessing and model architecture. Finally, in section 4, the results of the proposed model on the N-CMAPSS data set are presented.

## 2. PROBLEM DESCRIPTION

The model developed and tested in this paper was submitted for the 2021 PHM Society Data Challenge. The challenge description is thus the same as the problem description presented in this section.

### 2.1. Data Set Description

In this work, the new Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) data set is used to test a data-driven method for RUL estimation. This data set contains realistic run-to-failure data of turbofan engines (Chao et al., 2021). The N-CMAPSS data set offers higher fidelity data than the original CMAPSS data set by incorporating real recorded flight conditions and relating the degradation process to its operation history to extend the degradation model (Chao et al., 2021).

The data set models the failure trajectories of eight different failure modes that affect either the efficiency or flow of one or more of the sub-components of the engine. One of the failure modes included in the N-CMAPSS data set was not included in the 2021 PHM Society Data Challenge, so it was not used in the development of the tested model. The seven failure modes used in the data challenge are shown in Table 1. The failure modes span across all the rotating sub-components: fan, low-pressure compressor (LPC), high-pressure compressor (HPC), high-pressure turbine (HPT), and low-pressure turbine (LPT) and can affect either efficiency (E) or flow (F).

The flight durations are divided into three classes based on their length, but this work considers them together by using a window function, which is discussed in section 3. Each flight has an unknown initial condition and is run to failure, so the number of flights varies for each unit. Across all failure modes, the data set contained 90 units with data from a combined 6,825 flights. The data file for each flight contains scenario descriptors ($w$), measurements ($x_s$), a RUL label ($y$), and auxiliary data. The variables contained within the scenario descriptors, measurements, and auxiliary data are detailed in Table 2, 3, and 4, respectively.

### 2.2. Problem Definition

The goal of this work is to develop a model that estimates the RUL of a turbofan engine given a data set $\mathcal{D} = \{w_i, x_{s_i}, y_i\}_{i=1}^{N}$, which contains run-to-failure data for $N$ total flights of engines subject to different failure modes. The length of the sensory signals $w$ and $x_s$ is not constant between flights, so the model should be able to incorporate variable lengths of input data. The performance of the model is evaluated using a combination of the root-mean-square error (RMSE) and NASA's scoring function (Saxena, Goebel, Simon, & Eklund, 2008), calculated from the actual ($y$) and predicted ($\hat{y}$) RUL values. RMSE is calculated following Eq. (1) where $m_{v*}$ is the number of test samples. NASA's scoring function ($s_c$) is calculated using Eq. (2) where $\alpha$ is defined in Eq. (3). With $\alpha$ defined this way, under-estimations are preferable to over-estimations. The values calculated in Eqs. (1) and (2) are combined using Eq. (4) to get the final score.

$$RMSE = \sqrt{\frac{1}{m_{v*}} \sum_{j=1}^{m_{v*}} (y^{(j)} - \hat{y}^{(j)})^2} \tag{1}$$

$$s_c = \frac{1}{m_{v*}} \sum_{j=1}^{m_{v*}} exp(\alpha * |y^{(j)} - \hat{y}^{(j)}|) \tag{2}$$

$$\alpha = \begin{cases} \frac{1}{13} & \text{if} \quad y^{(j)} - \hat{y}^{(j)} \leq 0 \\ \frac{1}{10} & \text{if} \quad y^{(j)} - \hat{y}^{(j)} > 0 \end{cases} \tag{3}$$

$$score = 0.5 * RMSE + 0.5 * s_c \tag{4}$$

## 3. METHODOLOGY

### 3.1. Data Pre-Processing

In the data pre-processing phase, the scenario descriptors are combined with the measurements. This is done because the scenario descriptors can provide context to the measurement readings. For the time series data of a flight, which contains $m$ values, the scenario descriptors $w \in \mathbb{R}^{m \times 4}$ and measurements $x_s \in \mathbb{R}^{m \times 14}$ are combined to form $x \in \mathbb{R}^{m \times 18}$.

The flight duration is not consistent across each data point, so the length of each $x_s$ and $w$ varies. To account for this, a windowing function is used, which crops the measurement data and scenario descriptors to a specified length centered at

Table 1. Overview of the data set.

| Name | # Units | Fan | | LPC | | HPC | | HPT | | LPT | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | E | F | E | F | E | F | E | F | E | F |
| DS01 | 10 | | | | | | | ✓ | | | |
| DS03 | 15 | | | | | | | ✓ | | ✓ | ✓ |
| DS04 | 10 | ✓ | ✓ | | | | | | | | |
| DS05 | 10 | | | | | ✓ | ✓ | | | | |
| DS06 | 10 | | | ✓ | ✓ | ✓ | ✓ | | | | |
| DS07 | 10 | | | | | | | | | ✓ | ✓ |
| DS08 | 25 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 2. Scenario descriptors ($w$)

| Symbol | Description | Units |
|---|---|---|
| alt | Altitude | Units |
| Mach | Flight Mach number | - |
| TRA | Throttle-resolver angle | % |
| T2 | Total temperature at fan inlet | °R |

Table 3. Measurements ($x_s$)

| Symbol | Description | Units |
|---|---|---|
| Wf | Fuel flow | pps |
| Nf | Physical fan speed | rpm |
| Nc | Physical core speed | rpm |
| T24 | Total temperature at LPC outlet | °R |
| T30 | Total temperature at HPC outlet | °R |
| T48 | Total temperature at HPT outlet | °R |
| T50 | Total temperature at LPT outlet | °R |
| P15 | Total pressure in bypass-duct | psia |
| P2 | Total pressure at fan inlet | psia |
| P21 | Total pressure at fan outlet | psia |
| P24 | Total pressure at LPC outlet | psia |
| Ps30 | Total pressure at HPC outlet | psia |
| P40 | Total pressure at burner outlet | psia |
| P50 | Total pressure at LPT outlet | psia |

Table 4. Auxiliary data

| Symbol | Description | Units |
|---|---|---|
| unit | Unit number | - |
| cycle | Flight cycle number | - |
| Fc | Flight class | - |
| $h_s$ | Health state | - |

the middle of the flight. This ensures the length of the input data is consistent and has the additional effect of reducing the number of model parameters. The data is cropped about the middle instead of the beginning or the end because it is the most consistent across different flights and can lead to the most generalizable results. In this work, a window size of 32 is used, so after cropping $x \in \mathbb{R}^{32\times 18}$.

Finally, the input data is standardized so that it has zero mean and unit standard deviation. This done independently for each of the 18 features. The mean and standard deviation for each feature are calculated from the training data, then applied to both the test and validation data as:

$$X_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, \tag{5}$$

where $\mu_j$ and $\sigma_j$ are the mean and standard deviation of the $j^{th}$ feature, $x_{i,j}$ is the $i^{th}$ value of the $j^{th}$ feature, and $X_{i,j}$ is the final standardized value input to the model.

### 3.2. Model Architecture

The model used in this work is a deep convolutional neural network with an architecture inspired by the inception architecture originally proposed by Szegedy et al. (Szegeandy et al., 2015). The inception architecture makes use of inception modules, which contain parallel convolutional layers of different sizes and a single max pool layer. The outputs of each of these parallel layers are concatenated and sent to the next layer. The original inception module was developed for the classification of images and used two dimensional convolutions of sizes 1x1, 3x3, and 5x5 and a 3x3 max pool layer. The use of variable-sized convolutions allows for variable-sized features to be detected within a shallower neural network. This version of the inception module is shown in Figure 1a. To reduce the dimensionality of the model, convolutions of size 1x1 are incorporated before each convolution layer and after the max pool layer. An inception module that incorporates dimensionality reduction is shown in Figure 1b.

The inception model was developed for use on image classification, so the the convolutional layers within the inception modules were two dimensional. To adapt the model for use on the time series data from the N-CMAPSS data set, one-dimensional convolutional layers are used. The convolutions take place along the time axis.

The diagram of the network used in this work is shown in Figure 2. Two inception modules are used in the network. They are indicated in the figure with the dashed lines. In the first module, the dimensionality reduction before the convolution layers are not included so that the first convolutions see the raw signal. This is done because the input data is relatively small with 18 sensor readings, so the benefits of di-

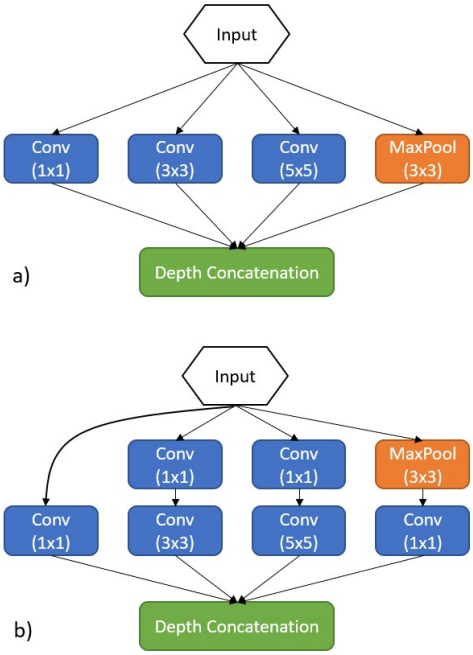


Figure 1. Inception module a) without dimensionality reduction and b) with it. Adapted from (Szegeandy et al., 2015).

mensionality reduction are minimal. In the second module, the dimensionality reduction is included because the stacked filters from the first inception module expand the dimensionality. After the second inception module, convolution filters are flattened and fully connected to a hidden layer with 256 cells. Dropout at a rate of 50% is included in this layer to reduce overfitting. Finally, in the last layer, a single node with a linear activation outputs the RUL estimation of the flight data under investigation.

In total, the model has 1.03 million trainable parameters. The breakdown of these parameters by layer is shown in Table 5. The number of parameters in each inception module is dictated by the number of filters in each element. A breakdown of the number of filters used in the inception modules is given in Table 6.

Table 5. Number of parameters broken down by layer.

| Layer | Output Size | # Params |
|---|---|---|
| Input | 30x18 | - |
| Inception1 | 30x108 | 5,940 |
| Inception2 | 30x128 | 37.4 k |
| Flatten | 3840 | - |
| Fully Connected | 256 | 983 k |
| Linear | 1 | 257 |
| Total | | 1.03 M |

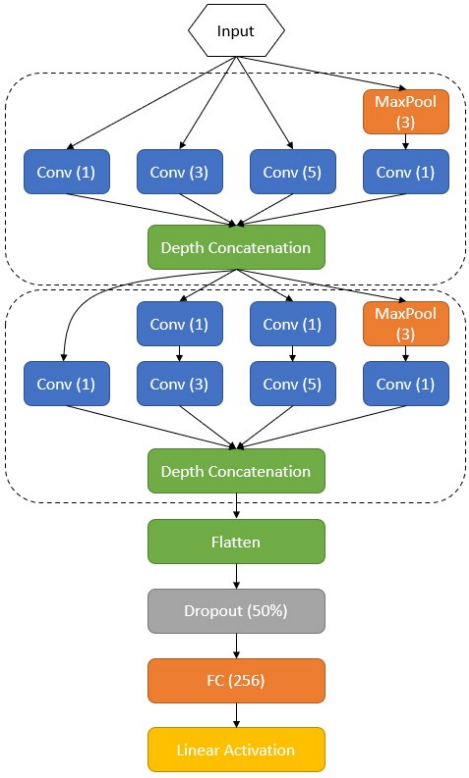


Figure 2. Model diagram

## 4. VALIDATION OF PROPOSED METHOD

### 4.1. Model Training

Across all the failure types and tested units, flight data from 6,825 flights were used for training and testing. Following the split of the data in the N-CMAPSS data set, 4,089 samples were used for training, and the remaining 2,736 were used for testing. Once the model was trained and tested using the public data set, it was used to classify the validation data set from the 2021 PHM Society Data Challenge. The model was implemented in Python using the Keras (Chollet et al., 2015) library and Tensorflow (Abadi et al., 2015) backend. In training, the accuracy of the model described above was evaluated by taking the RMSE between the predicted and actual remaining useful life of all the samples. RMSE was used in testing for ease of implementation, but score values as described in Eq. (4) were computed on the test data set. Training was conducted for 60 epocs using the Adams optimizer, a batch size of 64, and a learning rate of 0.001. Training was completed on an Intel Core i5 8th Gen CPU with 8 GB of RAM and took approximately five minutes.

### 4.2. Validation Results

The results of the trained model on the test data from the N-CMAPSS data set are shown in Figure 3 with the corresponding truth values, and the calculated RMSE, $s_c$, and score val-

Table 6. Number of filters within each inception module.

| Layer | # Conv (1) | # Conv (3) Reduce | # Conv (3) | # Conv (5) Reduce | # Conv (5) | MaxPool Reduce |
|---|---|---|---|---|---|---|
| Inception1 | 18 | - | 36 | - | 36 | 18 |
| Inception2 | 32 | 64 | 32 | 64 | 32 | 32 |

ues are shown in Table 7. The data in Figure 3 is sorted by the true RUL for ease of viewing.

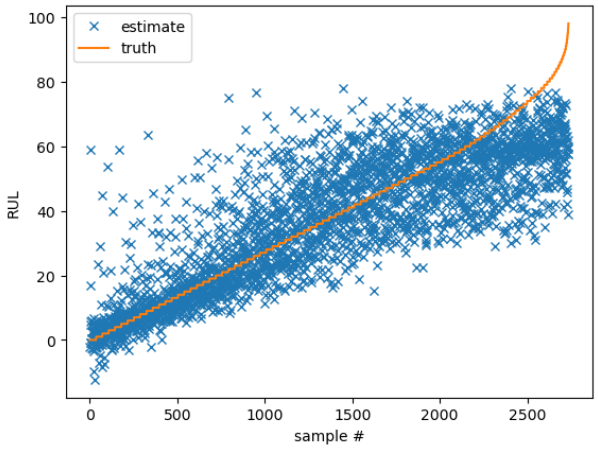


Figure 3. Model results on the N-CMAPSS test data set.

Table 7. Resulting error calculations for the prediction of the test data from the N-CMAPSS data set.

| Metric | Value |
|---|---|
| RMSE | 12.5 |
| $s_c$ | 2.53 |
| Score | 7.50 |

The RUL predictions on the first 14 units in the validation data set from the 2021 PHM Society Data Challenge are shown in Figure 4. The vertical lines indicate the transition from one unit to another. This figure demonstrates that the model does capture the decreasing RUL of each successive flight given only the previous flights' measurements. For the data challenge, the RUL prediction at the last sample in each unit was submitted. The resulting score on the validation data set was 3.33, which earned second place in the challenge.

## 5. CONCLUSION

In this work, a deep CNN architecture for the estimation of the RUL of turbofan engines based on measurements is proposed and tested on the N-CMAPSS data set. The proposed architecture incorporates inception modules with one-dimensional convolutions and only considers the readings from a single flight. The trained model achieved good accuracy on the N-CMAPSS test data and was further validated by earning the second place award for the 2021 PHM Society Data Challenge.

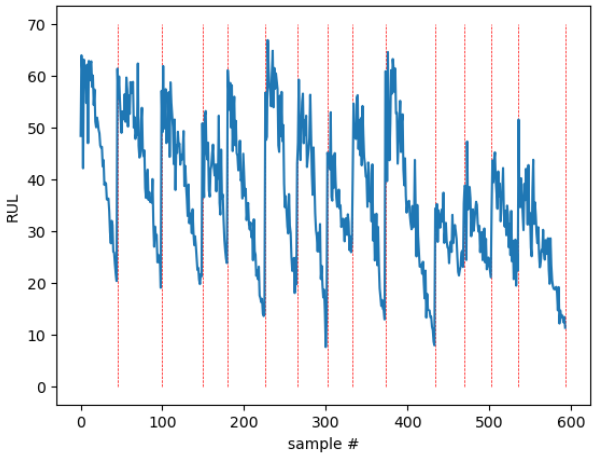


Figure 4. Model results on the first 14 units in the validation data set. Vertical lines indicate the transition from one unit to another.

The accuracy of the CNN developed here demonstrates that the information to make an accurate RUL prediction is contained within a single flight's measurements. This was demonstrated in previous work on the original CMAPSS data set, and its success here on the N-CMAPSS further validates its applicability. While accurate RUL estimations can be made using data from a single flight, the use of historical data may improve the accuracy of the RUL estimations. Future work should investigate incorporating historical data using recurrent architectures, such as LSTM.

One of the limitations of this work is that RUL estimations are not explainable, or the model is not able to explain why a RUL estimation was made. In future work, this could be improved upon by incorporating the ability of the model to predict what component is causing the RUL to be reduced. Since the N-CMAPSS data set contains multiple failure modes across all components, and the failure mode of each unit is labeled, future work on the N-CMAPSS data set should consider labeling the failing component, in addition to estimating the RUL.

## ACKNOWLEDGEMENTS

This work was supported by the U.S. Department of Energy DE-EE0008303 and the U.S. Department of Defense, Of-

fice of Local Defense Community Cooperation, Industry Resilience Program, Award #ST1449-21-03 2016-2166.

## REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from https://www.tensorflow.org/ (Software available from tensorflow.org)

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, 1–6. doi: 10.1109/ICEngTechnol.2017.8308186

Bolander, N., Qiu, H., Eklund, N., Hindle, E., & Rosenfeld, T. (2009). Physics-based remaining useful life prediction for aircraft engine bearing prognosis. *Annual Conference of the PHM Society*, *1*(1), 1–10.

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2021). Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA*. doi: 10.3390/data6010005

Chollet, F., et al. (2015). *Keras.* https://keras.io.

da Costa, P. R. d. O., Akeay, A., Zhang, Y., & Kaymak, U. (2019). Attention and Long Short-Term Memory Network for Remaining Useful Lifetime Predictions of Turbofan Engine Degradation. *International Journal of PHM Society*, *10*(4), 1–12. doi: 10.1115/GTINDIA2019-2368

Hochreiter, S., & Schmidhuber, J. (1997, nov). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. doi: 10.1016/j.ymssp.2005.09.012

Kong, H. B., Jo, S. H., Jung, J. H., Ha, J. M., Shin, Y. C., Yoon, H., ... Jeon, B. C. (2020). A hybrid approach of data-driven and physics-based methods for estimation and prediction of fatigue crack growth. *International Journal of Prognostics and Health Management*, *11*, 1–12. doi: 10.36001/ijphm.2020.v11i1.2605

Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering and System Safety*, *172*, 1–11. doi: 10.1016/j.ress.2017.11.021

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. *2008 International Conference on Prognostics and Health Management, PHM 2008*. doi: 10.1109/PHM.2008.4711414

Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H. (2011). Remaining useful life estimation - A review on the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14. doi: 10.1016/j.ejor.2010.11.018

Sikorska, J. Z., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, *25*(5), 1803–1836. doi: 10.1016/j.ymssp.2010.11.018

Szegeandy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June*, 1–9. doi: 10.1109/CVPR.2015.7298594

Wu, J., Hu, K., Cheng, Y., Zhu, H., Shao, X., & Wang, Y. (2020). Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. *ISA Transactions*, *97*, 241–250. doi: 10.1016/j.isatra.2019.07.004

Yang, H., Zhao, F., Jiang, G., Sun, Z., & Mei, X. (2019). A novel deep learning approach for machinery prognostics based on time windows. *Applied Sciences*, *9*(22), 4813. doi: 10.3390/app9224813

Zhao, C., Huang, X., Li, Y., & Iqbal, M. Y. (2020). A double-channel hybrid deep neural network based on CNN and BiLSTM for remaining useful life prediction. *Sensors (Switzerland)*, *20*(24), 1–15. doi: 10.3390/s20247109

Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long Short-Term Memory Network for Remaining Useful Life estimation. *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017*, 88–95. doi: 10.1109/ICPHM.2017.7998311