International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Predictive Maintenance of Gas Turbine using Prognosis Approach

Hima Soni[1], Akshay Kansara[2], Tanvi Joshi[3]

[1]Student of 4th Semester ME at S.P.B. Patel Engineering College (Saffrony), Ahmedabad-Mehsana Highway, Gujarat, India
[2]Head of the Department at S.P.B. Patel Engineering College (Saffrony), Ahmedabad-Mehsana Highway, Gujarat, India
[3]Assistant Professor at S.P.B. Patel Engineering College (Saffrony), Ahmedabad-Mehsana Highway, Gujarat, India

---***---

Abstract - *Industry 4.0 revolution aimed at transforming factories from automated to smart and intelligent. To fulfil that challenge, the research idea proposed herewith, implementing prognosis approach for Predictive Maintenance on Gas Turbine. Gas turbine (GT) based turbofan engines are recognized for their high availability and reliability and are used for aero, marine and power generation applications. Maintenance of this complex machinery should be done proactively to prevent premature failure, reduce the overall cost by avoiding unnecessary maintenance task. This goal is achieved by estimating the Remaining Useful Life (RUL) of GT. The RUL is very important information to decision-makers and planners for upcoming maintenance activity. This paper aims to explore the use of neural network models to predict RUL. In recent years researchers have proposed several machine learning, data driven and neural network approaches for predicting RUL. This paper investigates the effect of the Convolutional Neural Network (CNN) in RUL Estimation. The experimental study compares this approach to purely LSTM. This result suggests the CNN is a promising model in estimating RUL of time series dataset for GT even in the case of rare events.*

Key Words: remaining useful life, prognosis approach, gas turbine, predictive maintenance

1. INTRODUCTION

1.1 Gas Turbine

Gas turbine is the combustion engine. It is working by sucking air into the front of the engine using a fan. From there, the engine compresses the air, mixes fuel with it, ignites the fuel/air mixture, and shoots it out the back of the engine, creating thrust in case of turbofan engine [2] whereas in case of powerplant gas turbine it will generate the electrical energy. Two main applications of Gas Turbine engine are Turbofan engines and powerplant gas. Gas turbine-based Turbofan engine has been considered for the present research work. Gas turbine-based turbofan is one of the complex types of machinery, running 24/7 requires effective maintenance strategy to reduce downtime and for increased reliability and availability.

1.2 Types of Maintenance

The failure of GT engine is often a significant cause of major accidents and causalities. To prevent these causes, detecting primary degradation is essential. In the field of aircraft maintenance, traditional maintenance is either purely reactive (fixing or replacing an aircraft engine component after it's complete failure) or blindly proactive (assuming a certain level of performance degradation with no input from the aircraft engine itself and maintaining the aircraft engine on a routine schedule whether maintenance is needed or not). Both scenarios are quite wasteful and inefficient, and neither is conducted in real-time [7]. These kinds of Improper maintenance may lead to an increased rate of deterioration. Due to that maintenance cost may also reach up to 35% of operating cost [5]. Scheduling of maintenance activity based on fault diagnosis, performance degradation assessment and the predicted remaining useful life of the gas turbine and the need to prevent faults in advance, prognostics and health management (PHM) is gradually replacing these two maintenance strategies. [7]

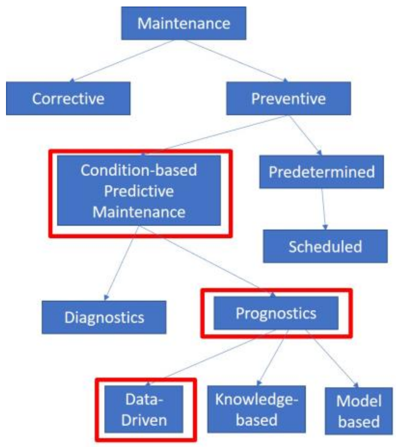


Fig-1: Types of Maintenance

1.3 Predictive Maintenance using IoT

Simple Predictive maintenance is a maintenance methodology that involves monitoring the health of a machine and applying predictive modelling techniques in order to predict the likelihood of failure of the machine and a time estimate of its probable occurrence [6]. Whereas, Predictive maintenance in the Internet of Things (IoT) era can be summarized as a maintenance methodology that brings together the power of machine learning and streaming sensor data to maintain machines before, they fail, optimize resources, and thereby reduce unplanned downtime. This is where IIoT helps in terms of providing all the data in the framework to make a meaningful assessment. [6]

1.4 Prognosis Approach

The main reason of sudden failures is that the diagnostics system cannot catch fault progression. To avoid such failures, the maintenance strategy needs to be changed from fail and fix to predict and prevent. In other words, instead of being reactive to be proactive. Prognostics is the main driver to proactivity [8]. Prognostics can be defined as a remaining useful life (RUL) estimation process of system/subsystem/component. In the beginning of life (BoL), the system operates normally with its full health. When the system starts to degrade, this can be considered as a trigger point to the prognostics system. The prognostics system continues to operate and performs RUL estimation at subsequent prediction points. In the same time, prognostics information is the main driver for condition-based maintenance (CBM) [8].

1.5 Estimating Remaining Useful Life (RUL)

There are three main classes of RUL prediction methods: (1.) data-driven methods, (2.) physics model-based methods, and (3.) methods that combine data-driven and physics model-based methods. The data-driven methods use past condition monitoring data, the current health status of the system, and data on the degradation of similar systems. There are two main challenges in prognostics based on physics: (1.) there is not enough physical knowledge to construct a physical degradation model and (2.) the values of the physical model's parameters are difficult to determine exactly. Therefore, it is important to understand the failure mechanism of the system correctly, and experienced personnel are required for physics-based models. Therefore, the requirements of data-driven methods to model the degradation and predict the RUL are easier to satisfy. [7]

The performance of many data-driven prognostics methods is heavily dependent on the choice of the performance degradation data to which they are applied. However, engines have many sensor parameters. The sensitivity of the data from different sensors varies in terms of showing engine performance degradation; the data from some sensors are sensitive and the data from other sensors are not sensitive. Therefore, it is necessary to select suitable sensor parameters whose data are more sensitive to the engine's performance degradation trend as the training data for the RUL prediction model. [7]

Three problems hinder the implementation of performance degradation feature extraction in practice. The traditional methods of extracting performance degradation features for prognostics are unsupervised and cannot automatically adjust the feature extraction modal parameters based on feedback from the prediction. Such feature extraction and choice are significant but represent a principal shortcoming of popular prognostics algorithms: the inability to extract and organize discriminatively or trend information from data. Therefore, it is important to develop an automatic feature extraction method that can extract the prominent feature to achieve better insight into the underlying performance degradation state. Deep learning, a new method that has been put forward in the last few years, can be used to extract multilevel features from data, which means the method could express data at different levels of abstraction. Deep learning is an end-to-end machine learning system.[7] Based on the present literature survey Convolutional Neural Network (CNN) is the one which is selected among all the Deep neural network approaches for the present research. CNN accepts the image as an input for that reason time series to image conversion is to be done. There are so many methods available to transform time series to images.

1.6 Imaging Timeseries

Recently, great results have been achieved by processing data with deep learning techniques, and, specifically, by using convolutional neural networks (CNN) with images as input. In scenarios where input data isn't formatted as an image, many transformation methods have helped apply CNNs to other data types. Time series is one of these data structures that can be modelled to approach the problem from a computer vision perspective.[28]

Recurrence plots are an advanced technique for visually representing multivariate non-linear data. This refers to a graph representing a matrix, where elements correspond to those times at which the data recurs to a certain state or phase. Recurrent behavior, such as periodicities or irregular cyclicities, is a fundamental property of deterministic dynamical systems, like non-linear or chaotic systems. As higher dimensional datasets can't be pictured easily, they can only be visualized by projection onto 2D or 3D sub-spaces. Recurrence plots enables the visualization of the mm-dimensional phase space through a two-dimensional representation of its recurrence. This recurrence of a certain state at time ii at a different time jj is marked within a 2D squared matrix and can be mathematically expressed as: [28]

$$R_{i,j} = \theta\,(\varepsilon_i - ||\,\overrightarrow{x_i} - \overrightarrow{x_j}\,||), \overrightarrow{x_i} \in R^m, \qquad i,j = 1,\ldots,N$$

The main advantage of using recurrence plots is being able to visually inspect any higher dimensional phase space trajectories by obtaining an image that hints at how the series evolve over time. [28]

2. PRIOR ART SEARCH

While proceeding with the research work, various approaches for the Predictive Maintenance using Prognosis approach have been analyzed in the prior art search. For predictive maintenance on Gas Turbine total, 4 research papers have been referred. As shown earlier predictive maintenance can be divided further in two parts. One is a diagnosis and other is the prognosis. Now prognosis can be done using Machine Learning approach, Data driven

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

approach or even using neural network. In the below table all the literatures are classified in different approaches.

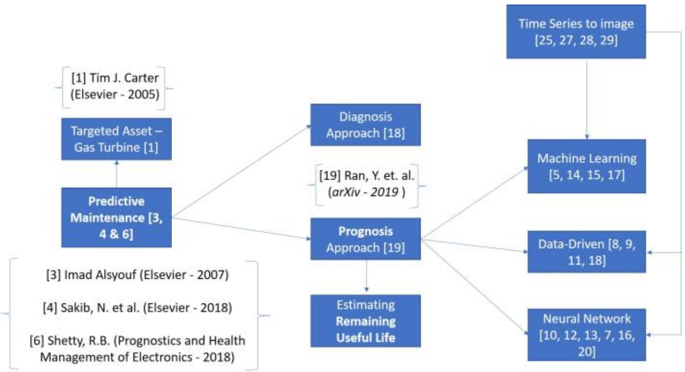


Fig-2: Taxonomy of Prior Art Search

Machine Learning	Data Driven	Neural Network
Autoregressive Model *Ahsan, S. et. al. (EDP Sciences - 2017) [8]*	**Data Driven Prognosis** *Elattar, H.M. et. al. (Springer - 2018) [11]*	**Directed Acyclic Graph Network Combined With CNN And LSTM** *Li, J. et. al. (IEEEAccess - 2019) [20]*
Bayesian Approaches *Mosallam, A. (Doctoral dissertation - 2014)[14]*	**Performance Based Health Monitoring** *Tahan, M. et. al. (Elsevier - 2017)[9]*	**Semi-supervised Deep Architecture** *Ellefsen, A.L. et. al. (Elsevier - 2019) [12]*
Data-level and Feature-level Fusion Ghorbani, S. et. al. (Springer - 2020) [17]	**Data-based Investigation On The Performance Of An Independent Gas Turbine** *Esan, A.B. et. al. (Elsevier - 2019) [8]*	**Multilayer Perceptron & Kalman Filter** *Alberto-Olivares, M. et. al. (IEEE - 2019) [13]* **Stacked Sparse Autoencoder With Multilayer Self-learning** *Ma, J. et. al. (Hindawi-Complexity - 2018) [7]*
Similarity Model & Support Vector Machine Approach *Chen, Z. et. al. (Energies 2018) [15]*	**Ensemble Of Data-driven Prognostic Algorithms** *Hu, C. et. al. (Elsevier - 2012) [18]*	**Hybrid Deep Neural Network** *Al-Dulaimi, A. et. al. (IEEE - 2019) [16]* **Transfer Learning** *Fan, Y. et. al. (arXiv - 2019) [10]*

Fig-3: Classification of Literatures in different Approaches for Prognostics

The purpose of this paper is to investigate the use of Deep learning-based models, to find remaining operational cycles (RUL) before failure in the test set in case of GT, which is the main objective of the present research work. As a supportive objectives Prognosis approach is used for performing predictive maintenance, Recurrence Plot method to transfer time series into images and Convolutional Neural Network has been selected to classify the data to find the remaining operational cycle.

3. DATASET DESCRIPTION

The Turbofan Engine Degradation Simulation Data Set by the Prognostic Center of Excellence of National Aeronautics and Space Administration (NASA) is used in the present research work. This data set was created by synthetic data collected from a thermodynamic simulation model called C-MAPSS (Commercial Modular Aero-Propulsion System Simulation). The simulator consists of 14 input parameters and 21 output parameters are reported in the data set. Each turbofan unit provides the following information: [13] The data set consists of multiple time series divided into 4 training and test subsets, both identified by the names: FD001, FD002, FD003 and FD004 (see fig-4). The whole data set contains sensor data from several turbofans, which operate normally at the beginning of the recording and eventually they develop a failure. [13].

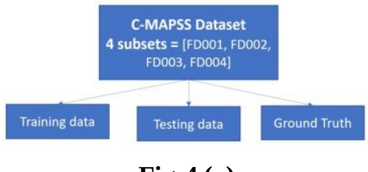


Fig-4 (a)

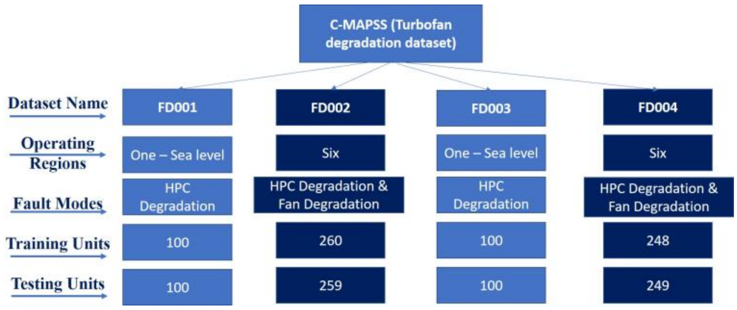


Fig-4 (b)

Fig-4: Dataset Description

There are two failure modes: high-pressure compressor degradation and fan degradation. As per the fig-4 (a).

4. METHODOLOGY

This section introduces the relevant procedure used in this research. As shown in Fig -5, the whole procedure for RUL prediction for a gas turbine-based turbofan engine consists of two main steps: data pre-processing and RUL prediction using CNN & LSTM.

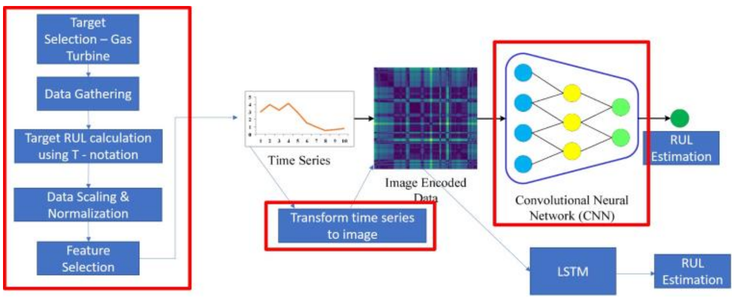


Fig-5: Overall Framework

Once the target asset selection is done, dataset must get prepared before applying Machine Learning Models. Data gathering, RUL calculation, data scaling and normalization and feature selection all these steps come under data preparation/preprocessing part. As CNN is only taking images as an input, so in this present research work timeseries data has been converted into the images. This will be one additional preprocessing step in case of CNN. At last training, testing and evaluation of the model will be held. Finally, RUL estimation shown as an output.

4.1 Data Gathering

Usually, the gas turbine-based turbofan engine is operating normally, at the starting of each time series and develops a fault at some point of time during the series. In the training set, the data recording ends when the turbofan stops working

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

completely due to the failure, representing a data base of turbofans that failed during operation. On the other hand, in the test set, the time series terminates sometime before the system stops working, representing a data base of turbofans in the current operation.

4.2 Data Preparation

Feature Selection, Data Normalization & RUL calculation are commonly coming under the Data Preparation step. Training & testing dataset will be pre-processed to achieve better results having good accuracy.

4.2.1 Feature Selection

Different sensors in gas turbine-based turbofan engine are having very different responses to the performance degradation process. Some sensors show unclear inclinations because of noise or insensitivity to degradation trends. Choosing unresponsive parameter data may reduce the RUL estimation accuracy. To improve the performance of the estimation model, sensors that are more responsive to the performance degradation process are chosen as inputs to the RUL estimation model.

4.2.2 Prepare Target columns

To train the model for RUL estimation, it is necessary to have a set of input and output data columns. Where the input data is the information recorded from several sensors and the output data is the RUL. However, databases for prognosis applications do not often contain the RUL information for training because, in many industrial applications, it is impossible to accurately assess the RUL information. Therefore, one should derive the RUL column (i.e. the time remaining before the end of each turbofan data recording), it can be derived as follows:

RUL Derivation:

$$LastCycle(Unitnumber) - CurrentCycle(i) = RemainingUsefulLife(RUL)$$

$$X - u_i = RUL, i = 1, \ldots, n$$

Let say, 'u' is unit number and each unit contain 'i' number of cycles. To calculate RUL, firstly the last cycle of each unit has been searched which denoted here as X. Likewise, RUL for each row will be calculated. In the case of regression, RUL column having continuous values is enough.

Labelling:

In the case of classification approach, one more column containing class labels needs to be derived. Which will be derived from RUL column.

For 0 to 15 remaining cycles, the given label is 2, 16 to 45 remaining cycles are labelled as 1 and for the RUL which are greater than 45 will be classified as 0. It is clear that in reality, the category labelled as 2 is the most economically valuable. Except for the convolutional neural network, data preparation part for other models are done up to this step.

4.2.3 Data Normalization:

After the elimination of some constant columns and selection of informative columns, the linear function (i.e. min-max normalization function) that best preserves the original performance degradation pattern of the aircraft engine is chosen to map the data for each selected sensor to [-1, 1].

4.3 Transform Time series to images

4.3.1 Recurrence Plot

As CNN model is being used to estimate the RUL, the transformation of whole preprocessed time series dataset into images are must be needed. That has been achieved with the help of the recurrence plot method.

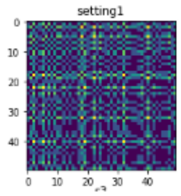


Fig-6: Sample Recurrence Plot of size 50x50x17

This image is an example of a recurrence plot which is nothing but the resultant representation.

To achieve the recurrence plot first extraction of the time series sequence is to be done.

4.3.2 Generating Sequence for transforming time-series into images

As an example, Unit 1 contains 192 cycles and consider window size as a 50. As a result of this generating sequence process, the total 142 sequences are generated of 50 window size as shown in fig-7.

Fig-7: Scenario of Timeseries Window Generation

Finally, at this stage preprocessing part for every Machine Learning model is completed.

4.4 Model Selection & Building

4.4.1 Model 1: CNN

Convolutional neural networks (CNNs) are a category of neural networks that have proven very effective in areas such as image recognition and classification. CNNs have been successful in identifying faces, objects, and traffic signs in addition to powering vision in robots and self-driving cars. CNNs derive their name from the "convolution" operator. The primary purpose of convolution in the case of CNNs is to extract features from the input image. the model learns how to automatically extract the features from the raw data that are directly useful for the problem being addressed. This is called "representation learning". The ability of CNNs to learn and automatically extract features from raw input data can be applied to time series forecasting problems. A sequence of observations can be treated like a one-dimensional image that a CNN model can read and refine into the most appropriate elements. This capacity of CNN has been proved to great effect on the time series classification task of turbo fan's remaining operational cycles prediction [26].

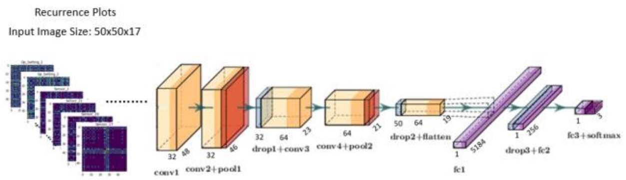


Fig-8: CNN Model Architecture

4.4.2 Model 2: LSTM

The Long Short-Term Memory network (LSTM) network, is a recurrent neural network that is trained using Backpropagation Through Time and overcomes the vanishing gradient problem. Instead of neurons, LSTM networks have memory blocks that are connected through layers. LSTM is capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods is practically their default behavior, not something they struggle to learn! All recurrent neural networks have the form of a chain of repeating modules of the neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer [35].

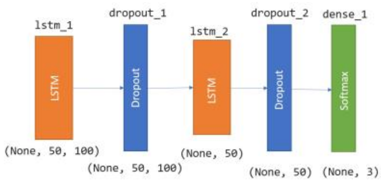


Fig-9: LSTM Model Architecture

4.5 Training, testing & validation

In this phase, by considering the randomly initialized parameters, training will be done. While training, cross-validation is done to avoid overfitting. Once the model is trained, parameters will get optimized for the next training and model will be retrained with the best parameters. Finally, the evaluation of the model will be held on the testing dataset.

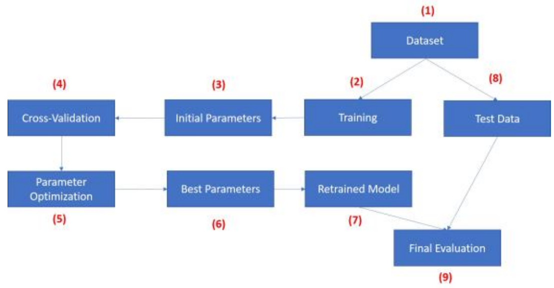


Fig-10: Flow of Training, Testing & Evaluation

4.6 RUL estimation

Finally, RUL will be estimated like, how much time (in terms of the number of cycles) is left before the next fault?

5. EXPERIMENTAL STUDY, RESULTS & DISCUSSION

All experiments are operated on Google Collaboratory platform having GPU. Using Colab individual can import an image dataset, train an image classifier upon it, and evaluate the model. Colab notebooks execute code on Google's cloud servers, that means one can leverage the power of Google hardware, including GPUs and TPUs. Also, Keras libraries are used having TensorFlow backend. Keras is a powerful open-source Python library for developing and evaluating deep learning models such as CNN and LSTM. It covers the efficient numerical computation libraries Theano and TensorFlow used for machine learning applications such as neural networks.

5.1 Understanding of the dataset through the plots

Once the CMAPSS dataset column labelling is done, training, testing and ground truth data sets are loaded. Data are available in the form of time series. Here in the below plot, one can conclude that unit 69 has the maximum number of cycles and unit 39 has the minimum number of cycles.

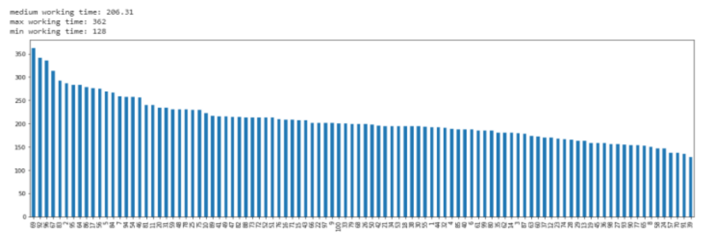


Fig-11: Plotting for the number of cycles of each unit

Just by plotting, identification of some data attribute containing constant data values can be done. Based on the identification, all the constant features have been eliminated in this step. To plot is always a good idea, in this way, one can have an impressive and general overview of the data at our disposal.

Fig-12: Plotting of Dataset Attributes

Verified feature selection with Correlation Matrix for all engine units. Out of all the columns these 17 columns are selected for further analysis: 'Op_Setting_1', 'Op_Setting_2', 'Sensor_2', 'Sensor_3', 'Sensor_4', 'Sensor_6', 'Sensor_7', 'Sensor_8', 'Sensor_9', 'Sensor_11', 'Sensor_12', 'Sensor_13', 'Sensor_14', 'Sensor_15', 'Sensor_17', 'Sensor_20', 'Sensor_21'

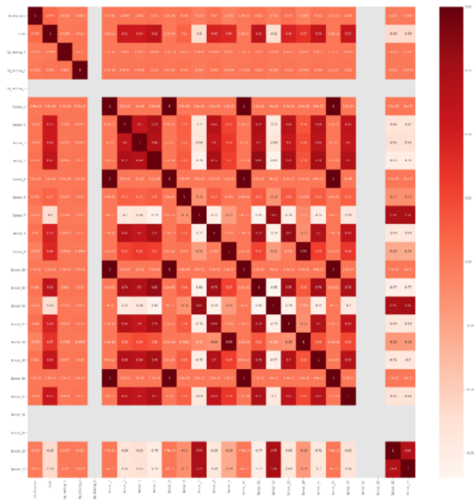
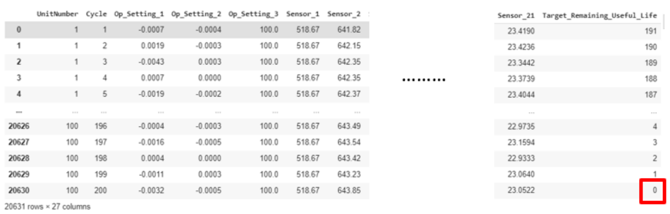


Fig-13: Correlation Matrix for Feature Selection

5.2 RUL Calculation of each row

This section calculates RUL in T-Minus notation. As covered in the methodology section, by finding the last cycle of each unit and then subtracting of a current cycle from it will results in RUL for each row. In case of training set, last cycle (time of failure) is provided in the data set itself.



	UnitNumber	Cycle	Op_Setting_1	Op_Setting_2	Op_Setting_3	Sensor_1	Sensor_2
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82
1	1	2	0.0019	-0.0003	100.0	518.67	642.15
2	1	3	-0.0043	0.0003	100.0	518.67	642.35
3	1	4	0.0007	0.0000	100.0	518.67	642.35
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37
...	...	...	...	...	...	...	...
20626	100	196	-0.0004	-0.0003	100.0	518.67	643.49
20627	100	197	-0.0016	-0.0005	100.0	518.67	643.54
20628	100	198	0.0004	0.0000	100.0	518.67	643.42
20629	100	199	-0.0011	0.0003	100.0	518.67	643.23
20630	100	200	-0.0032	-0.0005	100.0	518.67	643.85

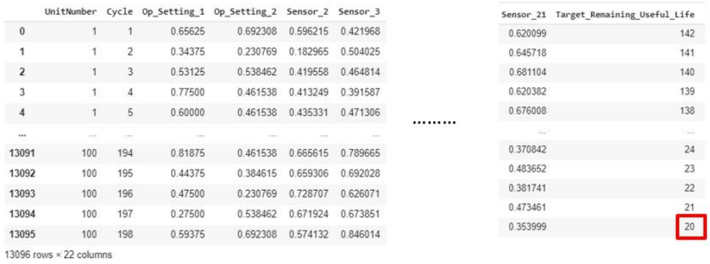
20631 rows × 27 columns

....

Sensor_21	Target_Remaining_Useful_Life
23.4190	191
23.4236	190
23.3442	189
23.3739	188
23.4044	187
...	...
22.9735	4
23.1594	3
22.9333	2
23.0640	1
23.0522	0

Fig-14: Snapshot of the training dataset

In case of test set the last cycle of each unit (i.e. time to failure(ttf)) is not given in the test set. So, to calculate the RUL of each raw, values given in the ground truth set will be considered as the value of the last cycle (ttf).



	UnitNumber	Cycle	Op_Setting_1	Op_Setting_2	Sensor_2	Sensor_3
0	1	1	0.65625	0.692308	0.596215	0.421968
1	1	2	0.34375	0.230769	0.182965	0.504025
2	1	3	0.53125	0.538462	0.419558	0.464814
3	1	4	0.77500	0.461538	0.413249	0.391587
4	1	5	0.60000	0.461538	0.435331	0.471306
...	...	...	...	...	...	...
13091	100	194	0.81875	0.461538	0.665615	0.789665
13092	100	195	0.44375	0.384615	0.659306	0.692028
13093	100	196	0.47500	0.230769	0.728707	0.626071
13094	100	197	0.27500	0.538462	0.671924	0.673851
13095	100	198	0.59375	0.692308	0.574132	0.846014

13096 rows × 22 columns

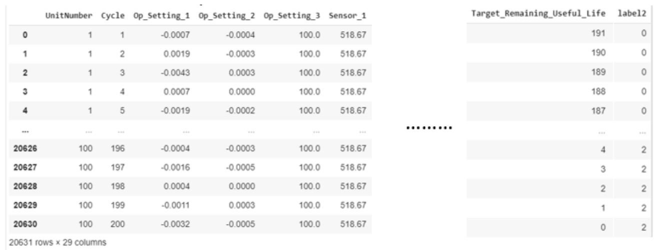
....

Sensor_21	Target_Remaining_Useful_Life
0.620099	142
0.645718	141
0.681104	140
0.620382	139
0.676008	138
...	...
0.370842	24
0.483652	23
0.381741	22
0.473461	21
0.353999	20

Fig-15: Snapshot of the testing dataset

5.3 Adding labels to the dataset

The class label has been given based on RUL column containing continuous values, as follows.



	UnitNumber	Cycle	Op_Setting_1	Op_Setting_2	Op_Setting_3	Sensor_1
0	1	1	-0.0007	-0.0004	100.0	518.67
1	1	2	0.0019	-0.0003	100.0	518.67
2	1	3	-0.0043	0.0003	100.0	518.67
3	1	4	0.0007	0.0000	100.0	518.67
4	1	5	-0.0019	-0.0002	100.0	518.67
...	...	...	...	...	...	...
20626	100	196	-0.0004	-0.0003	100.0	518.67
20627	100	197	-0.0016	-0.0005	100.0	518.67
20628	100	198	0.0004	0.0000	100.0	518.67
20629	100	199	-0.0011	0.0003	100.0	518.67
20630	100	200	-0.0032	-0.0005	100.0	518.67

20631 rows × 29 columns

....

Target_Remaining_Useful_Life	label2
191	0
190	0
189	0
188	0
187	0
...	...
4	2
3	2
2	2
1	2
0	2

Fig-16: Snapshot of Dataset after adding label column

5.4 Data Normalization

This step of data preparation is made using min-max pooling as discussed in the methodology section. Min-max normalization has been used to enable the unbiased contribution from the output of each sensor, i.e.,

$$xi = \frac{2(x_i - minx_i)}{max\, x_i - x_i} - 1,$$

where x_i is the time sequence of ith sensor measurements, and xi is the normalized sensor data. This normalization will guarantee equal contribution from all features across all operating conditions [34]. The normalized data will be between [-1,1].

5.5 Time Series to image transformation

gen_sequence() and gen_labels() functions are used to generate sequences from time series and label each window of size 50 respectively as shown in the below code snippet. For efficiency reason a 2D CNN requires spatial invariance. So, to transform the time series windows to images Recurrence Plots has been used. They are easy to implement in python Scipy with a few lines of code.

```
sequence_length = 50

def gen_sequence(id_df, seq_length, seq_cols):

    data_matrix = id_df[seq_cols].values
    num_elements = data_matrix.shape[0]
    for start, stop in zip(range(0, num_elements-
seq_length), range(seq_length, num_elements)):
        yield data_matrix[start:stop, :]

def gen_labels(id_df, seq_length, label):

    data_matrix = id_df[label].values

   num_elements = data_matrix.shape[0]
    return data_matrix[seq_length:num_elements, :]
```

5.6 Recurrence Plot

Using that function on time series sequence images are generated of size 50x50. One observation is made by an array of images of size 50x50x17. Where 17 is the number of non-zero variance columns.

```
def rec_plot(s, eps=0.10, steps=10):
    d = pdist(s[:,None])
    d = np.floor(d/eps)
    d[d>steps] = steps
    Z = squareform(d)
    return Z
```

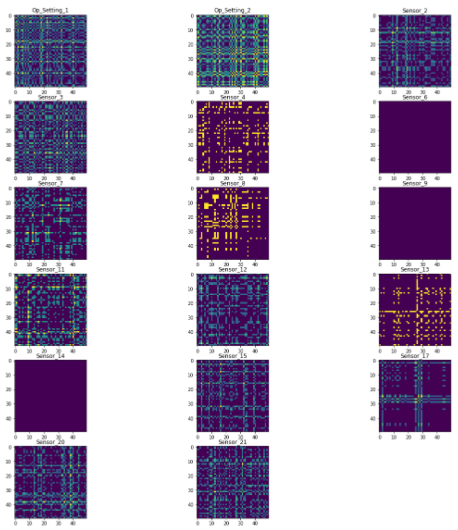


Fig -17: Snapshot of some observations

5.7 Model fitting & evaluation

5.7.1 CNN

Model is trained using 9 epochs using early stopping callbacks and to avoid overfitting cross-validation is applied. Adam optimizer is being used to tune and optimize the hyperparameter. Also, cross-validation is used to avoid overfitting. Finally using early stopping callbacks model get trained at 9th epoch only. Once model gets fitted now it's time to evaluate the model using a test set. While evaluating the trained model in case of FD001 dataset 91.74% accuracy is achieved.

5.7.2 LSTM:

Model is trained using 3 epochs using early stopping callbacks and to avoid overfitting cross-validation is applied. Once model gets fitted now it's time to evaluate the model using a test set. While evaluating the trained model in case of FD001 dataset 79.91% accuracy is achieved.

5.8 Results

In this sub-section, various experimental results have been presented to evaluate the performance of the CNN model for RUL estimation. Also, following the same steps, the model for FD002, FD003 & FD004 datasets has been trained and for every dataset, accuracy have been compared with LSTM model's resultant accuracy in the next subsection (5.9).

Table-1: Accuracy & Loss with CNN Model

Dataset Name	Accuracy	Loss
FD001	91.74%	23.28%
FD002	87.74%	36.33%
FD003	94.16%	19.24%
FD004	93.64%	39.50%

5.9 Comparison between CNN & LSTM

Table-2: Accuracy Comparison between CNN & LSTM

Accuracy comparisons Table				
Approaches used	**FD001**	**FD002**	**FD003**	**FD004**
CNN	91.74%	87.74%	94.16%	93.64%
LSTM	79.91%	80.50%	90.24%	74.18%

Implementing both models on the same preprocessed time-series data, it is found that CNN model is giving better accuracy than LSTM at the same time research objective has been fulfilled of classifying very rare events which are challenging to do with LSTM. CNNs are also good for feature extraction for the same reason, making them beneficial for transfer learning. For the system on which rarer events prediction is require in that way the decision has been made to go with CNN.

6. CONCLUSION AND FUTURE WORK

In this study, deep CNN and recurrence plot for the gas turbine-based turbofan engine have been explored. Segmentation of the time series dataset has been performed and generated recurrence plot image to train a deep CNN.

The system achieved accuracy in the range of 91.74% to 94.16% in benchmark dataset. Observation has been made that a deep CNN can learn recurrence plot from historical sensor data and can make a remaining useful life estimation. Using which one can achieve the following benefits,

1. Feature extraction is automatic.
2. In case of any prognostic health management dataset, one can train the applied CNN model with the use of user-friendly GUI and estimate RUL of the targeted assets, due to automatic feature extraction. Even though the user does not have much knowledge about the dataset and ML.
3. It will reduce the downtime as well as money loss. At the same time, it will increase the efficiency of the turbofan engine with the Just in Time maintenance.
4. Using Convolutional Neural Network one can classify the rare events using time-series data.

Overall, this work has demonstrated the performance of deep CNN to learn a recurrence plot pattern and estimate RUL. The future work will focus on improving the accuracy of the model. Besides, one can explore the performance to real-time data.

REFERENCES

[1] Rani, S., 2018. Common failures in gas turbine blade: A critical review. IJESRT (International Journal of Engineering Sciences and Research Technology, vol. 7, no. 3, pp. 799-803, 2018. doi: 10.5281/zenodo. 1207072.

[2] Boldmethod.com. (2019). How Does A Turbofan Engine Work? [online] Available at: https://www.boldmethod.com/learn-to-fly/aircraft-systems/how-does-a-jet-engine-turbofan-system-work-the-basics/.

[3] Alsyouf, I., 2007. The role of maintenance in improving companies' productivity and profitability. International Journal of production economics, 105(1), pp.70-78.

[4] Sakib, N. and Wuest, T., 2018. Challenges and Opportunities of Condition-based Predictive Maintenance: A Review. Procedia CIRP, 78, pp.267-272.

[5] Ahsan, S. and Lemma, T.A., 2017. Remaining Useful Life Prediction of Gas Turbine Engine using Autoregressive Model. In MATEC Web of Conferences (Vol. 131, p. 04014). EDP Sciences.

[6] Shetty, R.B., 2018. Predictive Maintenance in the IoT Era. Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things, pp.589-612.

[7] Ma, J., Su, H., Zhao, W.L. and Liu, B., 2018. Predicting the remaining useful life of an aircraft engine using a stacked sparse autoencoder with multilayer self-learning. Complexity, 2018.

[8] Esan, A.B., Ehiaguina, V., Awosope, C., Olatomiwa, L. and Egbune, D., 2019. Data-based investigation on the performance of an independent gas turbine for electricity generation using real power measurements and other closely related parameters. Data in brief, 26, p.104444.

[9] Tahan, M., Tsoutsanis, E., Muhammad, M. and Karim, Z.A., 2017. Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review. Applied energy, 198, pp.122-144.

[10] Fan, Y., Nowaczyk, S. and Rögnvaldsson, T., 2019. Transfer learning for remaining useful life prediction based on consensus self-organizing models. arXiv preprint arXiv:1909.07053.

[11] Elattar, H.M., Elminir, H.K. and Riad, A.M., 2018. Towards online data-driven prognostics system. Complex & Intelligent Systems, 4(4), pp.271-282.

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

[13] Alberto-Olivares, M., Gonzalez-Gutierrez, A., Tovar-Arriaga, S. and Gorrostieta-Hurtado, E., 2019, September. Remaining Useful Life Prediction for Turbofan based on a Multilayer Perceptron and Kalman Filter. In 2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) (pp. 1-6). IEEE.

[14] Mosallam, A., 2014. Remaining useful life estimation of critical components based on Bayesian Approaches (Doctoral dissertation).

[15] Chen, Z., Cao, S. and Mao, Z., 2018. Remaining useful life estimation of aircraft engines using a modified similarity and supporting vector machine (SVM) approach. Energies, 11(1), p.28.

[16] Al-Dulaimi, A., Zabihi, S., Asif, A. and Mohammadi, A., 2019, May. Hybrid Deep Neural Network Model for Remaining Useful Life Estimation. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3872-3876). IEEE.

[17] Ghorbani, S. and Salahshoor, K., 2020. Estimating Remaining Useful Life of Turbofan Engine Using Data-Level Fusion and Feature-Level Fusion. Journal of Failure Analysis and Prevention, 20(1), pp.323-332.

[18] Hu, C., Youn, B.D., Wang, P. and Yoon, J.T., 2012. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. Reliability Engineering & System Safety, 103, pp.120-135.

[19] Ran, Y., Zhou, X., Lin, P., Wen, Y. and Deng, R., 2019. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. arXiv preprint arXiv:1912.07383.

[20] Li, J., Li, X. and He, D., 2019. A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction. IEEE Access, 7, pp.75464-75475.

[21] Saxena, A., Goebel, K., Simon, D. and Eklund, N., 2008, October. Damage propagation modeling for aircraft engine run-to-failure simulation. In 2008 international conference on prognostics and health management (pp. 1-9). IEEE.

[22] Yang, C.L., Chen, Z.X. and Yang, C.Y., 2020. Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images. Sensors, 20(1), p.168.

[23] Hailesilassie, T., 2019. Financial Market Prediction Using Recurrence Plot and Convolutional Neural Network.

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

[25] Estebsari, A. and Rajabi, R., 2020. Single Residential Load Forecasting Using Deep Learning and Image Encoding Techniques. Electronics, 9(1), p.68.

[26] Lazzeri, F. (2019). 3 reasons to add deep learning to your time series toolkit. [online] O'Reilly Media. Available at: https://www.oreilly.com/content/3-reasons-to-add-deep-learning-to-your-time-series-toolkit/ [Accessed 27 May 2020].

[27] pyts.readthedocs.io. (n.d.). 4. Imaging time series — pyts 0.12.dev0 documentation. [online] Available at: https://pyts.readthedocs.io/en/latest/modules/image.html [Accessed 27 May 2020].

[28] Datascience.aero. (2019). From time to space! Time series converted into images to train CNNs. [online] Available at: https://datascience.aero/time-space-time-series-train-cnn/ [Accessed 27 May 2020].

[29] kaggle.com. (n.d.). RecuPlots and CNNs for time-series classification. [online] Available at: https://www.kaggle.com/tigurius/recuplots-and-cnns-for-time-series-classification [Accessed 27 May 2020].

[30] kaggle.com. (n.d.). Predictive Maintenance using LSTM on sensor data. [online] Available at: https://www.kaggle.com/nafisur/predictive-maintenance-using-lstm-on-sensor-data [Accessed 27 May 2020].

[31] Alhamaly, A. (2019). Jet Engine Remaining Useful Life (RUL) Prediction. [online] Medium. Available at: https://medium.com/@hamalyas_/jet-engine-remaining-useful-life-rul-prediction-a8989d52f194 [Accessed 27 May 2020].

[32] yajasd.github.io. (n.d.). Predicting Engine Failure using C-MAPSS data · Yajas Dwivedi. [online] Available at: https://yajasd.github.io/2018/06/04/Predicting-Engine-Failure/ [Accessed 27 May 2020].

[33] Cerliani, M. (2020). Remaining Life Estimation with Keras. [online] Medium. Available at: https://towardsdatascience.com/remaining-life-estimation-with-keras-2334514f9c61 [Accessed 27 May 2020].

[34] Ksenia Sorokina (2017). Image Classification with Convolutional Neural Networks. [online] Medium. Available at: https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8 [Accessed 15 Jun. 2019].

[35] Github.io. (2015). Understanding LSTM Networks -- colah's blog. [online] Available at: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.