Delft University of Technology

Adaptations for CNN-LSTM Network for Remaining Useful Life Prediction:

Adaptable Time Window and Sub-Network Training

Nick G. Borst

Adaptations for CNN-LSTM Network for Remaining Useful Life Prediction:

Adaptable Time Window and Sub-Network Training

by

N.G. Borst

in partial fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering,
specialisation Air Transport & Operations

at the Delft University of Technology,
faculty of Aerospace Engineering

to be defended on Monday Augustus 31, 2020 at 14:00.

Thesis committee:		
B.F. Santos	Chair	C & O: ATO
W.J.C. Verhagen	Responsible thesis supervisor	C & O: ATO
D. Zarouchas	Examiner	ASCM: SI&C

Cover image by Photo (c) izusek - Getty Images

Preface

The last six years were a terrific journey. Starting as a freshman in aerospace engineering I did not know what exactly was to come, however it contained the best moments of my life. In my first three years of the bachelor I learned a lot, however missed some practical experience. So I challenged myself to build and design the fasted solar powered boat as the technical manager of TU Delft Solar Boat Team. After this adventure I continued with my master in Air Transport and Operations with an internship at Damen Shipyards Group to use on board sensoring to predict operational behaviour of ships. Along the way I met many friends and enjoyed rowing a lot. This final thesis is my final step in this journey and end of my Aerospace Engineering study. Therefore, I would like to express my gratitude to everyone who has supported me during this part of my life.

Firstly, I want to thank my daily supervisor of the thesis, Wim Verhagen. For supervising me on this topic, even when meetings are harder to plan and perform due to the time difference between here and Australia. Also, the feedback and support helped me to continue working in these times where working from home is not always as easy. This helped me to deliver the final work as it is delivered here. I also want to thank Viswanath Dhanisetty for comments and tips during my kick-off, mid-term and green-light presentation and Marie Bieber for support and discussions on practical applications. I also would like to thank Bruno Santos and Dimitrios Zarouchas for joining the graduation committee and reading my work.

I also want to thank everybody involved in this great journey and the people that I met. I would like to thank my rowing team 'Steen Papier Bier', the solar boat 2018 team and everybody I worked with during my bachelor and master.
Finally, I want to thank my parents Gerard and Aukje and my girlfriend Selma. Ever grateful for all the support though the years.

My student life has come to an end and a new one begins. Thank you all.

Nick G. Borst

Delft, 12 August 2020

Contents

List of Figures

List of Tables

1
INTRODUCTION

New aircraft are equipped with a large amount of sensors and these sensors can be used to develop a variety of operational improvements in terms of scheduling and maintenance performance. One promising maintenance strategy, is to monitor the health of components and replace them before failure occurs. This is labelled as condition-based and/or predictive maintenance prognostics. This results in in higher safety levels, lower costs and improved up-time for aircraft. An important criteria for predictive maintenance is the Reliable Useful Life (RUL). The total amount of flight/cycles/-operations a given component can still perform before failure.
Recent research has shown that the use of deep learning techniques provide excellent results in predicting the RUL, especially the use of ensemble networks (a combination of different network types). The use of a combination of a Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) neural network proposes a promising method for predicting the RUL in a useful and accurate approach, which can be expanded and adapted to acquire a more accurate model. This results in the following main research aim:

This research aims to; reproduce, explain and further develop the CNN-LSTM model described by Li et al. [2] by using the CMAPSS data set.

This thesis is structured as follows. The main research, theory, experimental setup, methodology, results and conclusions are formulated in a scientific paper. This paper provides the main research of the complete thesis and can be read as a stand-alone document. Additionally, a total of six appendices are added, which contain all the work leading up to this research and additional information on specific topics of this research. In Appendix A and Appendix B the research methodology and the literature study are described. Appendix C describes the data pre-processing in more detail. Appendix D explains aspects of deep learning theory in more detail and the network structure is elaborated on. Appendix E provides additional figures for result interpretation and explains the adaptations in more detail. Finally, a description of the most used hyper-parameters are given in Appendix F as a reference.

2
PAPER

Not yet graded

Adaptations for CNN-LSTM Network for Remaining Useful Life Prediction:
Adaptable Time Window and Sub-Network Training

N.G. Borst (MSc. Student)
Section Air Transport and Operations, Department of Control and Operations,
Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands.

keywords: Prognostics, RUL prediction, Regression model, CNN, LSTM, Ensemble Method, CMAPSS

Abstract— **Estimating the RUL (Remaining Useful Life) of machinery is a useful tool for maintenance and performance operations. This results in lower costs, improved safety and operational improvements. This paper proposes two adaptations to the CNN-LSTM network provided by Li et al. [1], as well as exploring reproducibility, accuracy and sensitivity of the original DAG (Directed Acyclic Graph) network. The network at hand is an ensemble network combining LSTM and CNN neural networks to provide an accurate regression RUL prediction using the NASA CMAPSS dataset [2]. The Adaptable Time Window (ATW) adaptation increases the amount of time cycles that can be predicted and increases the accuracy, allowing for earlier predictions and better RUL predictions. Allowing state-of-the-art predictions accuracy for complex datasets. The Sub-network training adaptions did not surpass the accuracy of the original network with the current implementation settings, however is promising for further research.**

I. Introduction

Air traffic is growing worldwide, whilst more operational performance is required to keep up with the economic pressure. A high operational performance requires high up-time of aircraft. Therefore, an optimized way of performing maintenance is required to improve up-time, better scheduling and increased safety. Besides, maintenance contributes to 10 to 15 percent of the total airline costs [3]. This all results in a need for a better failure prediction method.

Current aircraft provide increasingly more on-board data generation. This data can be used for a vast variety of operational improvements. One of these improvements is to predict upcoming failures in advance. This can be achieved by a variety of techniques and algorithms. However, implementations of such algorithms are subject to a number of challenges: The number of failure events occurring is extremely low, due to a small amount of failure events. Sensor data and failure data should be combined in a single algorithm. The algorithm is required to be reliable and accurate to replace current maintenance measures. Finally, an adaptive and multi-functional algorithm is preferred to be applicable to more than a single system.

Maintenance can be divided in reactive maintenance and preventive maintenance. The latter is composed of three different types: Interval Based Maintenance (IBM), Condition-Based Maintenance (CBM) and Predictive Maintenance (PM) [4]. Predictive maintenance has recently gained more interest for predicting failure, since it can predict impending failure in advance and thus can improve the current maintenance system.

Recent research has shown great results in applying PM by using machine learning algorithms for predicting Remaining Useful Life (RUL). Especially applications of machine learning techniques such as tree-based Random Forrest (RF) and deep-learning Neural Networks (NN). These techniques mainly allow for a higher prediction accuracy, when used correctly. NN especially shows great results in terms of prediction capabilities. Although, this field is rapidly growing and different techniques are being researched at a high pace. Currently, a trend is showing, where versions of NN are used that combine different network types to achieve higher prediction accuracy by compensating for each other disadvantage. Also known as Ensemble Learning Methods (ELM). Another trend is to apply a regression model as an alternative for the more often used classification model. A classification model can only identify a certain state, while a regression model can predict any given state of the component. This regression can be leveraged to provide information during each cycle and enables real-time application. A RUL prediction can be provided for each cycle.

A combination of two different NN are showing superior results in determining RUL. This is a combination of a Long-Short Term Memory (LSTM) Network and a Convolution Neural Network (CNN). However, the ensemble and regression models are only applied quite recently and more research and improvements can be applied. This results in the following research aim: **This research aims to; reproduce, explain and further develop the model described by Li et al. [1] by using the CMAPSS dataset.** The two major improvements suggested by this paper are applying an Adaptable Time Window (ATW) and Sub-Networks Learning. With current Time Window operations, a RUL prediction cannot be made at the start of the algorithm initiation. Therefore, an Adaptable Time Window is applied to be able to predict a RUL in the early stages of the prediction. This method also improves the prediction accuracy at later stages, when more data is available. Most techniques aim to predict accurate results with the same network and settings for every data point. Others aim to predict the point where degradation is starting [5]. In this paper a Sub-Network method is described, which first identifies the stage and then uses a more specific trained network to improve the results.

As a side note, the reproducibility of many AI papers is quite challenging [6]. This papers aims to explain the used methods, concepts and parameters clearly to bridge the difficulty to apply such networks. More sensitivity analysis is applied for selection of certain parameters, where earlier reports did not further explore this domain.

The structure of the paper is as follows: In section II more information about previous research is given, as well as an overview of the theoretical background. In section III the data and conditions required are further explained. In section IV the techniques used by Li et al. [1] are described, as well as the improvements to this technique. The results are given in section V and finally the discussion and conclusion in section VII and section VI respectively.

II. Theoretical Background

This chapters provides an overview of earlier research and further information on the applied theory. With a focus on machine learning, deep learning, neural networks, CNN and LSTM algorithms. The following materials are recommended for further explanation about these topics: Goodfellow et al. [7], Russell et al. [8] and Sikoraksa et al. [9].

Earlier research can be classified in four categories: Knowledge based models, Data-driven models, Deep Learning and Physical models [9]. Knowledge based models are used to assess the current condition of a component based on previous failures. Although, these techniques require expert knowledge (set of rules made by an expert individual), are specific for each component and can result in contradictions (combinatorial explosion) [10]. A physical model can represent degradation by physically modelling the complete component. This is however, financially expensive and for some complex system currently not achievable. For example, crack life in metallic materials has been researched by Ray et al. [11].

Due to an increase in available data, new data-driven techniques could be applied. These data-driven approaches allow modelling of more complex systems, while no deeper understanding of the system is required. Early data-driven approaches are mainly based on statistical and stochastic models, such as Proportional Hazard Modelling, Static Bayesian Networks, Markov Models and Bayesian techniques with Kalman or Particle filters [9]. The main advantage of these techniques is that the results are statically interpretive and already applied for real-world prediction models. Still, stochastic models can be overly pessimistic for smaller datasets and assume that identical components are statistically identical (iid) and random variables are independent, which is often not true for datasets with random starting conditions, such as the CMAPSS dataset. Statistical models are more trend related and statistical interpretative, however artificial neural networks (ANN) often surpass their accuracy and with substantially more computational power in recent years, allow for more complex models [1]. Nonetheless, ANN operates as a black box model and is harder to interpret.

Machine learning and Deep learning are extremely large fields of science, which covers many different aspects of current research and is part of artificial intelligence. This besides caused the use of these algorithms for prognostic algorithms. These networks allowed more flexibility and a higher accuracy. The main disadvantage is that these methods have a wider confidence interval and requires a large amount of data and computational power. Another disadvantage is that NN models are a black-box model, certain features and choices cannot be statistically substantiated and harder to interpreted. However, these disadvantages can be overcome with correct application and adequate validation.

Early machine research focused on the use of flexible algorithms. For example Wu et al. [12] used a random Forest algorithm for tool wear and Majidian et al. [13] used a NN model to predict failure in boiler tubes. In 2008 a large aircraft turbofan dataset from the NASA repository was made available for a competition and further prognostic research, also known as the CMAPSS dataset [14]. This sparked new research in terms of classical data-driven methods and machine learning techniques. Babu et al. applied a CNN algorithm [15], followed by Zheng et al. to implement a LSTM algorithm [16] on the CMAPSS data. Finally, more ensemble techniques were applied with combination of CNN and LSTM. These showed better accuracy than earlier techniques [1, 17, 18].

The type of machine learning applied in this paper is a version of supervised regression machine learning. The model used is an ensemble method combining CNN and LSTM in a predictive regression model. The principles of a Deep Feed Forward Network, CNN and LSTM are further described in the next sections.

A. Deep Feed Forward Networks

There is a vast amount of different neural networks to apply. One of the simplest versions is a Deep Feed Forward Network (DFN), also called Multi-Layer perceptron's (MLP)[1]. A DFN consists of a number of nodes (neurons), which are connected in a directed acyclic graph. An example can be seen in Figure 1. Every node in a layer is connected to the previous layer and the next. When a feature input is given to the network, a value is given to each node. There are at least three layers required for deep learning. The first layers is called the input-layer, the middle layers are the hidden layers and the last layer is the output layer.

Learning of the algorithm is based on the principle of back-propagation [7]. The network is updated after a given batch input or epoch. This is achieved by comparing the

[1] Appendix D: Machine Learning and Network Methodology

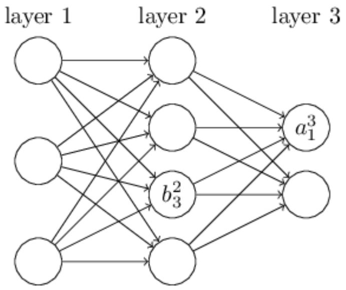


Fig. 1: Representation of a standard NN [19]

output of the network with the reference label (correct output). This process is repeated until a desired output is achieved, nevertheless overfitting can occur when too many training iterations are applied[2].

This results in a network, which gives an accurate prediction based on a set of input data. Every node in the network passes the data in a way that an accurate prediction is achieved.

B. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a specific type of neural network, where the input can be of a higher order of dimensionality. This technique originates from image recognition [20]. Every pixel in an image can be transferred to a grid of numbers based on their brightness. This results in a two dimensional grid of numbers, which represents the figure. A CNN can than evaluate these numbers and give a prediction on what the figure contains/is based on the earlier applied training procedure.

For this research a 2D-convolutional network is used which is based on sensor data readings and time. The data frame is further processed in a number of operations. Two types can be applied: convolution and pooling operations. A convolution operation places a kernel over the data. This kernel is moved over the data with a certain stride (direction of movement). A new matrix is built by the dot multiplications of the kernel with the input data. A depiction of this operation can be seen in Figure 2. A padding can be applied to the edges of the input data to maintain the matrix size after the operation. A pooling layer reduces the amount of data frames and the main features are highlighted, resulting in a faster and more accurate algorithm. It is similar to a convolutional operation, except that the pooling filter only applies a simple mathematical operation such as, max-pooling (highest value), average-pooling (average value) and others.

The main strength of a CNN is that features are extracted from the data, which cannot be defined by using standard pre-processing methods, such as statistics. It is especially practical when the data can be spatially ordered (in time or position).

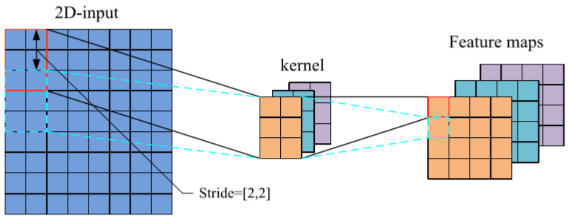


Fig. 2: CNN kernel operation [1]

C. Long-Short Term Memory Network and CNN-LSTM

An LSTM network is able to maintain information from a previous input values, by combining a short-term and long-term cell state. This technique allows for the network to use the complete data flow and alter predictions based on earlier inputs. The main strength of this technique is that later predictions become increasingly more accurate as time increases. Improved RUL prediction can be achieved, since aircraft component degradation is a continuous process. A LSTM network is highly applicable in situations where sequential prediction is required.

A CNN-LSTM is best used in a situation where a spatial and sequential input is available. This is the case of the CMAPSS dataset when certain pre-processing measures are used.

III. Experimental setup

This chapter comprises the used dataset, different pre-processing procedures and the performance metrics applied.

A. Dataset

The dataset used is the simulated C-MAPSS dataset, also known as the PHM08 Challenge Dataset, which is developed by NASA [2]. This data-set is created by simulating engine degradation and providing data for prognostic research. This dataset is divided in four different sub-datasets (FD001-004). Each dataset contains three different files: training file, testing file and a file with the actual RUL value for the final testing data points. The training and testing data both contain a number of engines. Each engine has a different initial health stage and different operation criteria thus provide a different number of operational cycles till failure. Each cycle of each engine contains operational settings and 21 sensor data values. The last cycle in the training set, also indicates failure afterwards. The last value of the testing set has a RUL equal to the value given in the file with the actual RUL.

Each sub-datasets has a different number of operating conditions and failure modes. The datasets are getting increasingly more complex. With FD001 having only one operating condition and fault mode, whilst FD004 has six operating conditions and two fault modes. The FD001 dataset is applied in this research, since application and optimization for all datasets would be time consuming and not the main topic of this research. An overview of the different data-sets can be seen in Table I .

[2] Appendix E: Training and Results

TABLE I: CMAPSS dataset overview

Parameter	FD001	FD002	FD003	FD004
Training Units	100	260	100	249
Testing Units	100	259	100	248
Operation condition	1	6	1	6
Fault modes	1	1	2	2

B. Pre-processing

Before feeding the data to the network, several steps are required[3]. First the data needs to be arranged per engine and the correct input features need to be selected. The selected sensors for this research are obtained by the procedure mentioned by Zheng et al. and Li et al.[1, 21]. The sensor that show no positive or negative trend over time (irregular) are not in the model. This results in a total of 14 sensors being used. Recent deep learning techniques are able to use a larger feature sizes, opposed to older non-deep learning algorithms, which use only a small amount of sensors [22]. The through put of the input is reduced When a sensor is showing no or less influence in RUL prediction for a deep learning network by changing the weights in the network. Older algorithms are more influenced by nonessential sensors.

The next step is to normalize the data, since neural networks require this to work optimally, without resulting in an exploding gradient [7]. The chosen methods of normalization are Z-score normalization and min-max normalization. Z-score handles outliers well, in contrast to min-max normalization. However, Z-score does not maintain its exact scale, while min-max does.

An often used pre-processing technique is a piece-wise linear RUL function, instead of a linear RUL function. This was introduced by Heimes [23]. The maximum target RUL is limited with this technique. This allows the model to represent real life degradation more accurately. It is based on the principle that after a certain amount of time degradation is visible and during normal operation no failure is apparent. A representation of piece-wise linear target function can be seen in Figure 7.

Another technique applied is originating from Zhao et al. [24]. A Time Window (TW) over the data points is created with this technique. The window is created by sliding over the training data for a certain TW length , step size of one and the label being the last cycle of every window. This allows the use of CNN and the ability to implement a representation over time. A baseline TW of 30 is applied for the FD001 dataset, as described by Li et al. for the baseline [1]. An adaptable TW technique is described in subsection IV-B, which allows for early prediction and enhanced prediction for longer sustaining components. Finally, an overview of the pre-processing parameters is given in Table II.

TABLE II: Data pre-processing and training parameters

Pre-processing		Training	
Parameters	Value	Parameter	Value
Time Window	30	Epochs	10
Selected sensors	14	Training samples	17731 (for TW30)
Number of slices	10	Mini-batch size	100
Normalization	Score	Optimizer	RMS-Prop
RUL target	Piece-wise Lin.	Learning rate	0.001
Max RUL	125	Loss function	Smooth L1 Loss
		Activation function	ReLu
		Deep learning software	Pytorch/Python

C. Performance metrics

The used performance metrics are Root Mean Square Error (RMSE) and a scoring function, first introduced at the 8th international prognostics and health management conference [25]. This method penalizes both early and late prediction errors equally. The equation for RMSE is given in Equation 1, with N being the test sample size and h_j the prediction error.

$$\text{RMSE} = \sqrt{\frac{1}{N} \cdot \sum_{j=1}^{N} h_j^2} \tag{1}$$

The scoring function penalizes late predictions more than early failures predictions. Predicting early failures is for maintenance predictions far less critical than late predictions. The score can be calculated using Equation 2 with N the test sample size, h_j being the prediction error. Equation 2 is modified with respect to the original function to also accommodate score over the complete dataset, instead of only the last prediction. s_j is divided by Testing Samples (TS) and multiplied by the number of Testing Units (TU). This allows the accuracy ,when only looking at the last sample, to be compared with the complete set.

$$\text{Score} \quad = \sum_{j=1}^{N} \frac{s_j}{TS} \times TU, \quad s_j = \begin{cases} \mathrm{c}^{-\frac{h_j}{13}} - 1, & h_j < 0 \\ \mathrm{e}^{\frac{h_j}{10}} - 1, & h_j \geq 0 \end{cases} \tag{2}$$

The performance can be calculated based on the complete test set, as well as the final value, which is given for the CMAPSS test dataset. The final value is the RUL prediction of the last available data point for each individual engine unit. Most authors use the latter to evaluate the effectiveness of their algorithm, since this was the main goal of the PHM08 challenge. [1, 17, 26]. Nonetheless, prediction accuracy over the whole dataset can be applied to see its overall effectiveness and is more realistic when real-time application is available.

In the result section both RMSE and Score are provided. Also the division is made between the complete set and only the final RUL. This results in a total of four different accuracy metrics.

IV. Method

The model based on the research done by Li et al. [1] is used in this paper. This model bases itself on an ensemble network based on a parallel adaptation of a 2D CNN network and a LSTM network. The outputs of both networks are afterwards combined in a second LSTM network. A visual representation of this network can be seen in Figure 3. A

[3] Appendix C: Data pre-processing

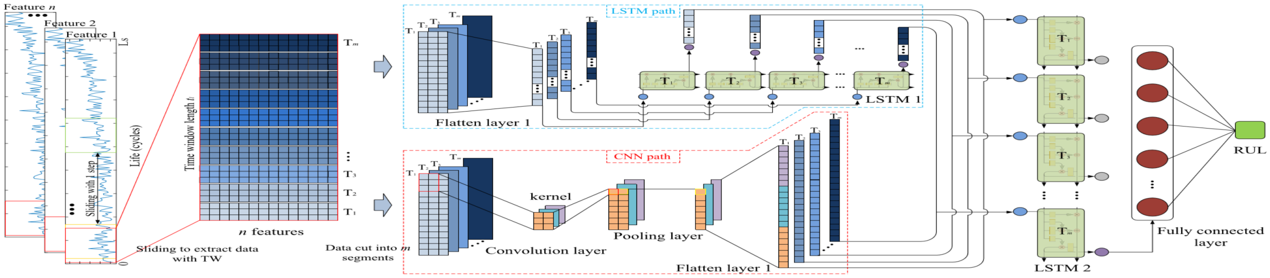


Fig. 3: A representation of the network architecture [1]

TABLE III: Dag Network parameters for FD001 (* is variable)

Symbol	Definition	Value
tl	Time window length	30/*
n	Number of sensors	14
BS	Batch Size	100
Ls	Life span	*
m	pieces	10
nk	Number of kernels	3 k1
Kernel size	2	
k2	Kernel size	2
s1	Stride	2
s2	Stride	2
p1	Pooling size	2
p2	Pooling size	2
ps1	Pooling stride	2
ps2	Pooling stride	2
u1	LSTM nodes	18
u2	LSTM nodes	10

description of the symbols used can be found in Table III. This is followed by two adaptations of the algorithm: the Adaptable Time Window and Sub-Networks training.

A. DAG Network

The network used is a directed Acyclic Graph Network (DAG). The input data is pre-processed as described in subsection III-B. This results in an input data size of a $BS \times [t_l \times n]$. This data is then transposed and cut into m pieces. Resulting in a data size of $BS \times [n \times (t_l/m)] \times m$. This data is then implemented into two different parallel paths: LSTM path and CNN path.

LSTM path: The input data first has to flattened since a LSTM network cannot directly use higher-order dimensional data. This flattening results in a data frame of $BS \times (n \times t_l/t_n) \times m$. The LSTM network contains u_1 nodes and one layer. The outputs size of a LSTM network is equal to the number nodes. So, the final output is equal to $BS \times u_1 \times m$.

CNN path: The other path is based on a CNN algorithm. First a convolutional operation is implemented over the input data. This operation has a kernel of size $[k_1, k_2]$, three different kernels (n_k) and a stride of $[s_1, s_2]$. Finally the data is flattened $(6 \times nk)$ to be able to combine both paths. This results in output of $BS \times u_1 \times m$, which is the same as the LSTM path.

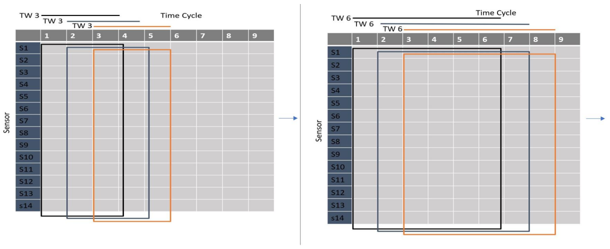


Fig. 4: Overview of the application of ATW for TW 3 and TW 6. Left a TW of 3 is applied for the time cycles 3,4 and 5. Right a TW of 6 is applied for the time cycles 6,7 and 8.

Combining: To obtain a final outcome, the CNN and LSTM paths are combined. First the output of both paths are summed element-wise. This data is then further implemented in a second LSTM network. This network consists of u_2 nodes. The output of this network is $BS \times m \times u2$. Only the last output of this network will be input to a fully connected layer. This fully connected layer has a single output for each prediction. This output is the final estimated RUL of size BS.

B. Adaptable Time Window

Most techniques that use a Time Window for prediction apply this at a cost of no prediction during the first number of cycles [1, 17]. This has no implication when only predicting the last cycle of each unit in the testing set. However, this is undesirable for real-time application, hence the following method is suggested.

The network is trained for different types of Time Windows using all available samples. This is applied in increments of 3 (e.g. 3,6,9 etc.), since the convolutional operations are applicable for these increments due to the size of slices. Other increment steps could be achieved when the size of slices is altered. The weights of each training are stored and later used in the application of the model. Longer TW steps lead to less available samples, due to required data points to create a sample of this size.
Output of the model can then be taken at any time cycle by taking the algorithm weights for largest possible TW (e.g. for time cycle 6,7 and 8 you can use a TW of 6). An overview of the different sample sizes can be seen in Figure 4. The process is repeated for a TW length up to 45.

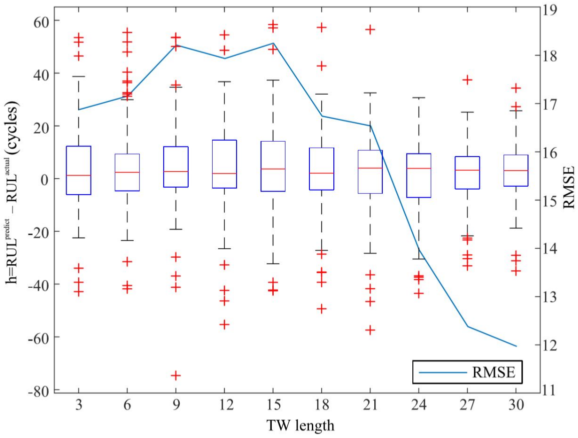


Fig. 5: RMSE Prediction error of the last prediction of each engine of different Time Window lengths[1]

C. Sub-network training

The second adaptation to the network is based on sub-networks. First, a prediction is made by the original trained network (primary network). Afterwards a second network (sub-network) can be applied, which is only trained on the given health stage samples. The RUL prediction of sub-networks is used for the final prediction. This results in a double regression model. An overview of the procedure can be seen Figure 7.

By training in specific stages the accuracy of the network can be increased. A global network predicts less accurate, since all different inputs should result in the correct outcome. However, when the primary network specifies a data sample in the wrong sub-network (incorrect classification) an error is introduced. This error is introduced by classifying the sample in the incorrect sub-network. The introduced is required to be lower than the increase in accuracy by higher accuracy of the given sub-networks.

The original data samples are divided in three stages. The healthy stage, degradation stage and critical stage. These stages are constructed by the separation of two lines. The line, which divides the healthy and the degradation stage is named upper boundary (UB) and the line between the degradation stage and the critical stage as the Lower boundary (LB). The locations of these boundaries can be varied for optimal accuracy.

The first prediction is achieved by using the network parameters and weights for the complete datasets[4]. The sub-networks are trained separately by training with only samples that are labelled with a RUL in a given stage. This results in three differently trained networks, which can be later applied to the testing data. The test data is divided in the three stages and feed to the specific trained networks (sub-network). The outcome of this sub-network is used as the outcome of the complete network. A representation of the health stages is given in Figure 6.

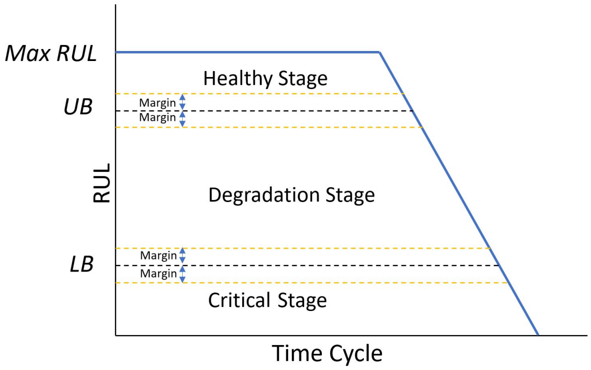


Fig. 6: Representation of the different health stages for sub-network training

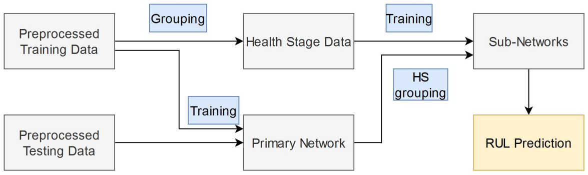


Fig. 7: Overview of the sub-network training procedure

A margin is introduced around the UB and the LB. When a prediction of the primary network is between the margins, no sub-network is applied. The primary prediction is used as the outcome for that sample. This method reduces the effect of incorrect classification and different margin sizes are applied for optimal accuracy.

D. Training

The network requires training to be able to predict an accurate RUL for a given input. This training is performed over a number of iterations, known as epochs. The required number of epochs is based on the type of dataset, learning rate and batch-size applied. The selection of these parameters is important for the prevention of overfitting the model. An overview of the training parameters used for the model are depicted in Table II. The training data is firstly divided in a mini-batch of 100, as also applied by Li et al. [1]. This mini-batch allows the network to be updated after a sample of 100 data frames. During each epoch, the network is fed with all the different mini-batches. After each mini-batch, the network is updated with the given learning rate. This process prevents overfitting by only providing the network with a small amount of frames and outperforms other training methods: updating after the complete set and after every single input (Batch gradient descent and Stochastic gradient descent) [27]. Mini-batch training allows the network to be trained with longer sets of data without overfitting on long term data relations. However, applying a mini-batch is computational more costly than batch gradient descent, since the weights need to be updated more often.

After each mini-batch the weights need to be tuned. For this the type of Activation Function (AF), loss function, learning rate and optimizer are important. The AF used

[4] Appendix F: Hyper-parameters

is the Rectified linear unit (ReLu) AF. A commonly used activation function, which is computationally faster than the original Sigmoid AF and more consistent than Leaky ReLU, Parametric ReLU or Swish . However, some might outperform ReLu in certain prediction scenarios [28]. The loss function applied is the Smooth L1 Loss function (Huber Loss). This type of loss is mostly applied for regression problems and is suitable in most cases. Exploding of the gradient prevented more often and it is less sensitive to outliers compared to mean squared error loss [29]. The Learning Rate (LR) indicates the quantity each weight is updated each mini-batch. A low LR can find the optimal point of minimum more easily than a higher LR, however the training might converge to a local minimum. A higher LR convergences faster to an optimum, however might not be able to find the optimal point. The LR is varied after the suitable normalization type and optimizer are chosen. After the learning is chosen, a suitable stopping point is also allocated (amount of epochs to train). When trained for too long, the network can only recognize the training data itself due to overfitting. When too few training cycles are applied, underfitting is applicable. An optimal amount of epochs is required.

A suitable optimizer is required for good training. An optimizer indicates the direction each weight is to altered after each update. The most commonly used optimizers are Standard Gradient Descent (SGD) with momentum, RMS-Prop and Adam optimizers. SGD with momentum is able to find more flatter local minima, however tuning of the weights is more important and critical. RMS-prop adapts the LR automatically when converging to a minimum. Adam is a combination of both techniques and also possesses an adaptable LR. These techniques are therefore selected and both tested for the best accuracy. [27]

V. Results

The results provided are based on 10/20 iterations. The accuracy of median iterations is presented based on the testing set. Besides, the stability of the network can be visualised (e.g. how accurate is the network when randomly trained). The performance metrics given are the RMSE, Final RMSE, Score and Final Score.

First, the results of the primary DAG network are explained, followed by the two adaptations.

A. DAG Network

The DAG network used is initialized with the parameters as specified in section IV. However, many training and pre-processing parameters can be tuned. However, the optimizer and normalization type are often shortly explained in literature and no further optimization criterion are provided [1, 17, 18]. First the normalization and optimizer are chosen by training the network for 20 iterations. An example can be seen in Figure 9 and overview of the different metrics are shown in Table IV. These also include the results of the best performing network. From these results can be seen is that the RMS-Prop optimizer in combination with the Z-score normalization display the best results for all performance metrics. Nonetheless, when looking at the best performance values for the Last RMSE and Last Score are the best accuracy is achieved for the min-max normalization. Z-Score normalization also converges faster and thus requires less epochs, which is computationally less expensive.

The RMS optimizer is for most accuracy metrics outperforming the Adam optimizer, which is unexpected, since Adam is a combination of RMS-prop and SGD-momentum optimizers. However, the Adam optimizer is not increasing the accuracy for this model. Therefore Adam optimizer is discarded for further trade-offs.

Secondly, for determining the training parameters a variation of initial learning rates (LR) are applied for Z-score and min-max normalization. The RMS-opt optimizer changes the learning afterwards according to the updates. An overview of the different learning rates can be seen in Table V. The optimal epoch is given in between the square brackets. For this epoch, the given metric accuracy has the lowest median value. A higher initial LR does converge faster, however an optimal point is not found. This is especially true for the higher learning rates. A lower initial LR does converge better, however takes longer to converge and can get stuck in an un-optimal local minimum.

Z-score normalization is however more accurate, especially when the complete testing set is used. The learning rate 0.001 is showing accurate results and converges at a small amount of epochs, which increases computational time. Therefore Z-score optimization with a LR of 0.001 is selected for further analyses. The training sequence can be seen in Figure 8.

However, the optimal RMSE is higher than the RMSE achieved by Li et al., which is 11.96. This probably has to do with some minor network settings, which are very hard to reproduce and are not stated (such as loss-function, CNN activation function etc.). The selected amount epochs is also not specified in this paper (stopping point). Nonetheless, the values of Table V will be used in the rest of the report as a benchmark for further adaptations.

In Figure 10 a representation of the RUL prediction is depicted along with the actual target RUL. This shows the predictability of the model over the complete life span of a single unit. It is based on the earlier specified parameters, normalization, learning rate and optimizer. The first 29 time cycles have no prediction, since the Time Window length requires that sample space. Afterwards, the prediction is given until almost failure. What can be seen is that early predictions and late predictions are more accurate than for the middle cycles. This behaviour can also be seen in Table VIII and is most likely due to the harder to predict tipping points between healthy and degradation. Degradation is as prominently measurable at these points.

Finally the height of the maximum RUL is varied and the results can be seen in Table VI. Two additional accuracy

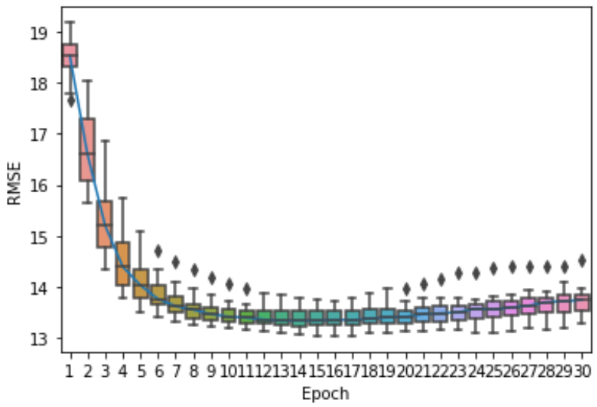


Fig. 8: RMSE of the RUL prediction of the selected network configuration with a LR of 0.001

TABLE IV: Overview of the median accuracy regarding optimizers (20 iterations)

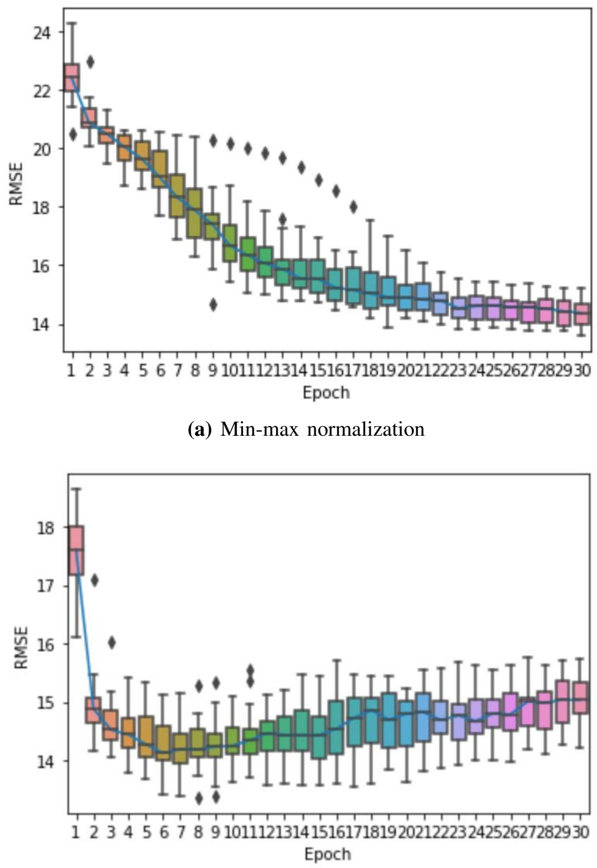
Operation	RMSE	Final RMSE	Score	Final Score
Z-Score normalization				
Adam (median)	14.13	14.66	449.34	440.64
Adam (minimum)	13.37	13.38	333.28	319.86
RMS-Prop (median)	**14.06**	**13.39**	**370.96**	**319.14**
RMS-Prop (minimum)	13.16	12.60	307.65	254.68
Min-max normalization				
Adam (median)	14.37	14.75	443.28	409.67
Adam (minimum)	13.60	13.65	358.51	312.98
RMS-Prop (median)	15.05	14.11	433.38	370.67
RMS-Prop (minimum)	13.98	12.43	331.55	248.95

metrics are added for this table, which represents the final RUL and score, with respect to the unmodified target label with the RUL-target function. These metrics are named the Actual Final RMSE and Score.

What can be seen in the data is that a lower maximum RUL results in a higher accuracy for the complete data set and for only the final samples. However, the ability to predict the original label is decreasing when lower maximum are applied. In the actual label a number samples have a RUL of >120. In practice a lower maximum RUL increases the accuracy, however it reduces the ability to predict further in advance. The choice of the maximum RUL depends on the maintenance operation, component and planning.

TABLE V: Overview of the median accuracy regarding learning rate (20 iterations)

Learning rate	RMSE	Final RMSE	Score	Final Score
Z-score norm.				
0.005	14.13 [5]	14.66 [4]	449.34 [8]	440.64 [6]
0.0025	13.63 [10]	13.25 [5]	346.52 [6]	310.44 [16]
0.001	13.34 [14]	13.73 [10]	351.07 [10]	369.24 [23]
0.0005	13.36 [21]	13.71 [22]	346.85 [16]	353.35 [15]
0.00025	13.50 [7]	14.09 [26]	317.86 [4]	388.58 [23]
Min-max norm.				
0.005	15.05 [28]	14.11 [26]	433.38 [28]	370.67 [22]
0.0025	14.40 [27]	13.78 [28]	390.44 [29]	320.02 [27]
0.001	14.44 [29]	13.91 [27]	414.10 [29]	343.87 [28]
0.0005	14.57 [29]	14.16 [28]	454.01 [29]	364.18 [27]
0.00025	15.65 [29]	14.91 [29]	515.13 [29]	400.11 [29]



(a) Min-max normalization

(b) Z-score normalization

Fig. 9: RMSE of the DAG network with a LR of 0.005 and Adam optimizer

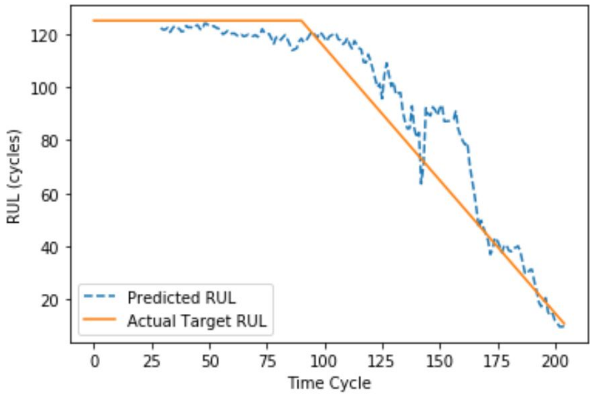


Fig. 10: RUL prediction for test engine unit 76 (RMSE 11.1, Score 222.9) using the parameters as specified in Table II and Table III

TABLE VI: Overview of the median accuracy regarding Maximum RUL (20 iterations)

Maximum RUL	RMSE	Final RMSE	Actual Final RMSE	Score	Final Score	Actual Final Score
linear	36.01	26.16	26.16	3.71e4	4830.87	4830.87
135	15.64	14.76	15.00	491.74	426.19	434.16
125	13.34	13.74	15.07	351.07	369.24	406.01
115	13.59	11.87	15.04	341.70	234.35	331.72
105	12.58	9.57	16.32	280.93	147.66	339.00
95	11.81	7.92	19.4	233.24	94.11	510.48

B. Adaptable Time Window

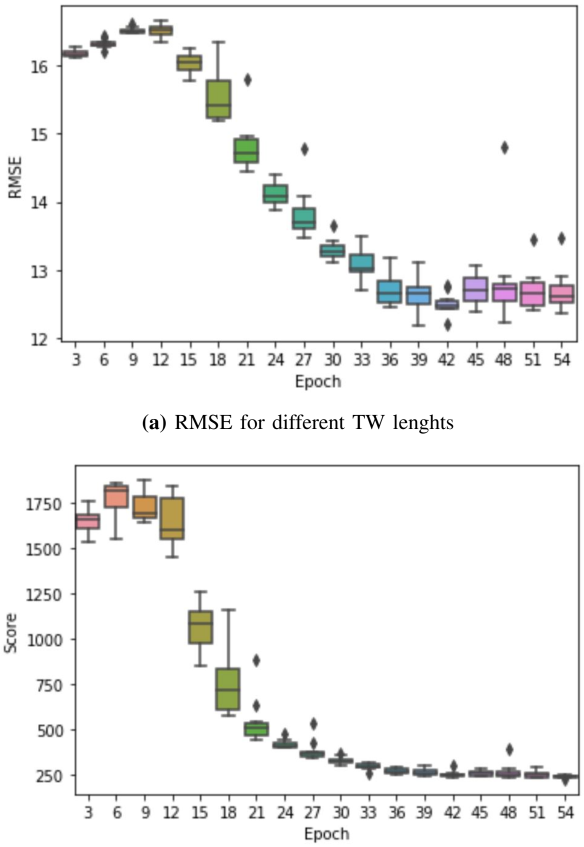
The networks is trained for different Time Window lengths. An overview of all the different accuracies with the different Time Window lengths can be seen in Figure 11. A total of 10 iterations are performed for each TW length and the box plot shows the results of these 10 iterations. The results for the first 30 TW length frames are similar to Figure 5. The score also shows similar behaviour compared to the RMSE, where the error increases and then decreases to a certain optimum.
After TW 30 the accuracy increases with a bigger time window. The RMSE has an optimum at 42 epochs, whilst the score keeps on decreasing even further. This shows that an increase in Time Window length, increases the accuracy. The two main reasons that the accuracy is increasing whilst also adding time cycles with a worse accuracy is that most early cycles are commonly in the healthy stage, where a better accuracy is already present (see also Table VIII). Another reason is that there are more points added with the longer time window than there are points in the early cycles.

In subsection IV-B the accuracies are shown for three different types of ATW. The network used for each different TW length, is the one where it's given accuracy metric is median of all training iterations. This takes into account the relative randomness of neural network training. The first type of ATW takes the highest possible TW length for each time cycle. This results in an improvement for all the different metrics (e.g. time cycle 12,13 and 14 use a TW length of 12). Two other types are introduced, which are based on the effect that the time cycles, where the RUL prediction is low, a TW of 3 is optimal. ATW 11 and ATW 14 predicting time cycles up to 11 and 14 respectively. What can be seen is that the accuracy increases even further, since early time cycles are more accurate to predict with a TW length of 3. The difference between ATW 11 and ATW 14 for predicting the RUL for the final time cycles of each engine is extremely small, since very few final cycles are below 15. However when taking all predictions into account a difference is present, since early cycles are present.

Early cycle predictions are also possible with this technique. The standard version of the DAG network with a Time Window of 30 is only able to predict cycles from 30 cycles and onward, which is unpreferable for real applications.

TABLE VII: Overview of accuracy regarding ATW (10 iterations)

Learning rate	RMSE	Final RMSE	Score	Final Score
Reference	13.34	13.73	351.07	369.24
ATW	11.17	12.77	179.23	268.26
ATW 11	11.09	12.75	176.69	267.17
ATW 14	12.75	11.09	177.38	267.14



(a) RMSE for different TW lenghts

(b) Score for different TW lenghts

Fig. 11: Median accuracy for different Time Window lengths (10 iterations)

C. Sub-Networks

The results for a lower bound of 30 and an upper bound of 100 are displayed in Table VIII. The accuracy of each different health stage is shown for two different models. The 'normal' reference DAG network is applied on each different health stage, as well as the sub-network training. The 'normal' DAG network uses the results of the primary network and does not calculate a new value for the sub-network. The accuracies of the sub network training are based on the location where each sample is

TABLE VIII: Overview of the median prediction accuracy regarding sub-network training with a LB of 30 and an UB of 100 (10 iterations)

Network	RMSE	Final RMSE	Score	Final Score
Ref.	13.34	13.73	351.07	369.24
All health stages	13.31	13.85	449.13	562.21
Critical	5.34	4.80	49.08	38.94
Critical ref.	5.27	4.46	53.78	46.03
Degradation	17.90	13.85	600.26	314.65
Degradation ref.	18.80	20.23	818.10	792.84
Healthy	11.31	14.46	281.57	393.72
Healthy ref.	10.64	9.87	254.7	123.78

TABLE IX: Overview of the prediction accuracy regarding sub-networks error types

Bounds	A [%]	B [%]	C [%]	D [%]
LB30 & UB100	5.1	34.3	1.6	25.5
LB50 & UB110	10.5	33.4	2.5	69.0

indicated by the primary network. The original label could be used, however the accuracy would be influenced by other sub-network settings.

What can be seen is that the network using all the health stage sub-networks is hardly improving in terms of RMSE. The other accuracy metrics are even showing inferior behaviour. This likely is associated with the effect that the primary network wrongly classifies samples in the incorrect health stage. In Table IX an overview can be seen of four different types of miss-classification. A represents the samples that should be placed in the healthy life stage, but are placed in the degradation stage. B and D should be placed in the healthy or critical stage respectively, however are placed in the degradation stage. Finally C, should be placed in the critical stage, but it is placed in the degradation stage.

The different learned sub-networks based on the different health stages show that degradation stage prediction is increased in accuracy. However, the other health stages are performing less accurate with respect to the reference. As a side note can be seen that for the reference primary network is not as effective in identifying the RUL during the degradation life phase. This might be useful for further research to improve this section of the prediction.

The number of different errors show that most of the errors occur in B and D. This would indicate that increasing the UB and LB should reduce these errors. However, when these bounds are increased the type of errors is still rising. For example an UB and a LB of 50 and 110 provides a testing RMSE and score of 13.45 and 571.56 respectively.

Another approach, which could be applied is to create margin around each boundary. The effect of different margins are shown In Table X. However, an increase in the margin does not improve the prediction accuracy.

VI. Discussion

The DAG network provided by Li et al. possesses an excellent accuracy and is reproducible to a certain degree of accuracy. Nonetheless, finding the right parameters and constructing the correct network is a time consuming practice. Many parameters can be tuned, where the outcome of certain choices can only be evaluated by the outcomes of the network. Many parameters where chosen by selection on their characteristics. However, some were tested on different settings.

TABLE X: RMSE prediction accuracy with a given margin

Margin	RMSE
0	13.31
2	13.71
5	13.98
10	14.47

The first adaptation on the network provided improved results. All the different metric types were better performing than its original counterpart. The accuracy of other earlier applied techniques can be seen in Table XI. The Adaptable Time Window DAG 14 network currently surpasses the accuracy of earlier stated models.

This method could especially be applied for real-life applications. A RUL prediction is already available from the third time cycle of a unit/component. Accuracy is increasing when more time cycles are available. The results of this model could be used for planning maintenance in advance and updates regarding the health status of the component are updated with each cycle completed. The next step is to implement this algorithm on datasets where data is available for the complete operational profile (not only a single value per sensor per cycle). For example using the data per flight phase or even each second.

The sub-network technique is not yet increasing the accuracy of the network. Although, for future research the following steps could be taken. Firstly, the data could be re-normalized after the samples are divided by the primary network. This results in a larger difference and might yield better results. An optimization of different bounds can be applied. Also, more or less different boundaries might be of positive influence. A technique could be defined, which detects falsely placed samples and relocates them to the correct health stage sub-network. A combination of these techniques might result in an improvement with respect to the original DAG network.

Furthermore, the use of Pytorch and Python is highly recommended for research purposes. Tensorflow is another widely used deep learning library. The most important difference is the way the graph and network setting are initiated. In Pytorch the graph is dynamic and can be modified during training. In Tensorflow the graph is static and needs to be properly initiated beforehand. This is especially practical when different sizes of inputs are available for LSTM/RNN models. In this example, the TW is kept constant during the complete training, however during training this could be modified. This is more useful for sound and word recognition. Another advantage of Pytorch is that every specific network setting can be changed. For example each layer can bet set with a different activation function. This can be leveraged to achieve the best

TABLE XI: Prediction RMSE of other models (See Li et al. for references to the given models [1])

Methods	Years	RMSE Last
MLP	2016	37.56
SVR	2016	20.96
RVR	2016	23.80
CNN	2016	18.45
LSTM	2017	16.14
ELM	2017	17.27
DBN	2017	15.21
MODBNE	2017	15.04
BLSTM	2018	14.26
RNN	2018	13.44
DCNN	2018	12.61
BiLSTM	2018	13.65
DAG	2019	11.96
ATW DAG	2020	11.09

network optimization. Other recommendations for further work consist of the following. Firstly, this network or similar networks should be applied to real-life complex data-sets. This is to evaluate the application of RUL prognostics in reality. Another recommendation is to replace the LSTM with GRU (Gated Recurrent Unit) and compare the results. A GRU is a less complex version of a LSTM network, however, might outperform it on certain smaller Time Window lengths.

VII. Conclusions

The model proposed by Li et al. was successfully recreated, however the direct results of the network were slightly worse than the accuracies stated by Li et al. This is probably due to minor network settings, which can make differences in the results. A set of different parameters, which were not further elaborated on by earlier research were tested. A combination of RMS-prop optimization, Z-score normalization and an initial learning rate of 0.001 were selected as the best performing parameters with a RMSE and score of 13.34 and 351.07, respectively. The maximum RUL for the target function should be lower for a higher accuracy, however ability to predict longer RUL predictions reduces. Reducing the maximum RUL improves the accuracy of the network, however the ability to predict more in advance is also reduced. Selection depends on the required RUL in practice.

When applying an Adaptable Time Window (ATW) technique the prediction accuracy increases to a RMSE of 11.09 and a Score of 176.69. This is due to the effect that later predictions are predicted more accurately and early predictions are added, which are naturally better predicted. This technique allows RUL predictions already from the third time cycle.

The Sub-network training technique was also successfully applied. However, the results were not yet showing improvement over the reference DAG network. This probably has to do with classifications of the primary network and the overall accuracy of the network themselves. Applying a margin around the boundaries of each sub-network did not improve the accuracy for this situation.

The DAG network provided is a well performing regression model, which is able to provide accurate RUL prediction from the third life cycle onward, which matches current state-of-the-art techniques. The stability of the network is provided by introducing a number of iterations per training.

References

[1] J. Li, X. Li, and D. He, “A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction,” *IEEE Access*, vol. 7, pp. 75464–75475, 2019.

[2] Nasa, “Nasa Ames prognostic data reprository,”

[3] IATA, “Airline Maintenance Cost: Executive Commentary Edition 2019 (FY2018 data),” *URL: http://www. iata. org/whatwedo/ . . .*, no. January, pp. 1–16, 2019.

[4] R. K. Mobley, *An introduction to predictive maintenance*. Butterworth-Heinemann, 2002.

[5] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, and S. Poll, “Echo State Network for the Remaining Useful Life Prediction of a Turbofan Engine,” *European Conference of the Prognostics and Health Management Society 2016*, pp. 1–15, 2016.

[6] O. E. Gundersen and S. Kjensmo, “State of the art: Reproducibility in artificial intelligence,” *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 1644–1651, 2018.

[7] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning,” tech. rep.

[8] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[9] J. Sikorska, M. Hodkiewicz, and L. Ma, “Prognostic modelling options for remaining useful life estimation by industry,” *Mechanical Systems and Signal Processing*, vol. 25, pp. 1803–1836, 7 2011.

[10] A. K. Garga, K. T. McClintic, R. L. Campbell, Chih-Chung Yang, M. S. Lebold, T. A. Hay, and C. S. Byington, “Hybrid reasoning for prognostic learning in CBM systems,” in *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, vol. 6, pp. 2957–2969, 3 2001.

[11] A. Ray and S. Tangirala, “Stochastic modeling of fatigue crack dynamics for on-line failure prognostics,” *IEEE Transactions on Control Systems Technology*, vol. 4, pp. 443–451, 7 1996.

[12] D. Wu, C. Jennings, J. Terpenny, R. X. Gao, and S. Kumara, “A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests,” *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 139, 7 2017.

[13] A. Majidian and M. H. Saidi, “Comparison of Fuzzy logic and Neural Network in life prediction of boiler tubes,” *International Journal of Fatigue*, vol. 29, pp. 489–498, 3 2007.

[14] E. Ramasso and A. Saxena, “Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset,” *PHM 2014 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014*, pp. 612–622, 2014.

[15] G. Sateesh Babu, P. Zhao, and X.-L. Li, “Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life,” in *Database Systems for Advanced Applications* (S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, and H. Xiong, eds.), (Cham), pp. 214–228, Springer International Publishing, 2016.

[16] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, “Long Short-Term Memory Network for Remaining Useful Life estimation,” *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017*, pp. 88–95, 2017.

[17] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, “HYBRID DEEP NEURAL NETWORK MODEL FOR REMAINING USEFUL LIFE ESTIMATION Electrical and Computer Engineering , Concordia University , Montreal , QC , Canada H3J 1P8 Concordia Institute for Information System Engineering (CIISE), Concordia University , Cana,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3872–3876, 2019.

[18] L. Jayasinghe, T. Samarasinghe, C. Yuenv, J. C. Ni Low, and S. Sam Ge, “Temporal convolutional memory networks for remaining useful life estimation of industrial machinery,” *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2019-Febru, pp. 915–920, 2019.

[19] M. A. Nielsen, *Neural Networks and Deep Learning*. 2015.

[20] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” 5 2015.

[21] C. Zheng, W. Liu, B. Chen, D. Gao, Y. Cheng, Y. Yang, X. Zhang, S. Li, Z. Huang, and J. Peng, “A Data-driven Approach for Remaining Useful Life Prediction of Aircraft Engines,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-Novem, pp. 184–189, 2018.

[22] T. Wang, J. Yu, D. Siegel, and J. Lee, “2008 INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems,” tech. rep., 2008.

[23] F. O. Heimes, “2008 INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT Recurrent Neural Networks for Remaining Useful Life Estimation,” tech. rep., 2008.

[24] Z. Zhao, Bin Liang, X. Wang, and W. Lu, “Remaining useful life prediction of aircraft engine based on degradation pattern learning,” *Reliability Engineering and System Safety*, vol. 164, no. 457, pp. 74–83, 2017.

[25] A. Saxena and K. Goebel, “Phm08 challenge data set,” in *NASA Ames Prognostics Data Repository*, NASA Ames Research Center, 2008.

[26] V. Mathew, T. Toby, V. Singh, B. Maheswara Rao, and M. Goutham Kumar, “Prediction of Remaining Useful Lifetime (RUL) of Turbofan Engine using Machine Learning,” *Proceedings of 2017 IEEE International Conference on Circuits and Systems (ICCS 201*, 2017.

[27] S. Ruder, “An overview of gradient descent optimization algorithms,” pp. 1–14, 2016.

[28] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for Activation Functions,” 10 2017.

[29] P. Jha, “A Brief Overview of Loss Functions in Pytorch,” *Medium*, 2019.

A
Research Methologies

Previously graded under under AE4010

Developing prognostics for Boeing 787 system regarding Reliable Useful Life (RUL) predictions

By Nick G. Borst, 4370457
C & O, ATO

A.1. Executive Summary

Newer aircraft are producing more and more data, which can be used for predicting failure behaviour. However, there are several challenges in aircraft maintenance prediction such as few failure events, many different data sources, lack of practical applications and reliable algorithms. Literature research showed that the newest form of maintenance managment is based on predictive maintenance, being one step more advanced than condition-based maintenance. The time till failure is being analysed instead of anomalies in the data.

For Reliable Useful Life (RUL) prediction, four types of models are available. From this, the most potent model type is the machine learning branch. Machine learning consists of many algorithms and mostly require large amount data. New research shows great performance with the use of neural network techniques. The most promising being the Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM) networks. However academic research is often only based on experimental data, instead of real-life data. Industry currently relies mostly on statistical models to predict if a certain maintenance action is required in advance. However, some steps are taken to implement condition-based maintenance with less advanced machine learning techniques.

The main goal of the project is formulated as: To achieve an accurate and reliable RUL prediction of the Boeing 787 CACTS system by means of applying a Neural Network algorithm. This leads to the following research question: Is it possible to predict the Reliable Useful Life of the Boeing CACTS system by using a Neural Network using on-board generated data? To full-fill this research question a CNN-LSTM network will be used to predict the RUL. This requires data filtering, model set-up, parameter optimization and training.

This research contributes to the understanding of RUL prediction with real data and aims to develop a predictive maintenance algorithm, which is accurate, reliable and applicable in practice.

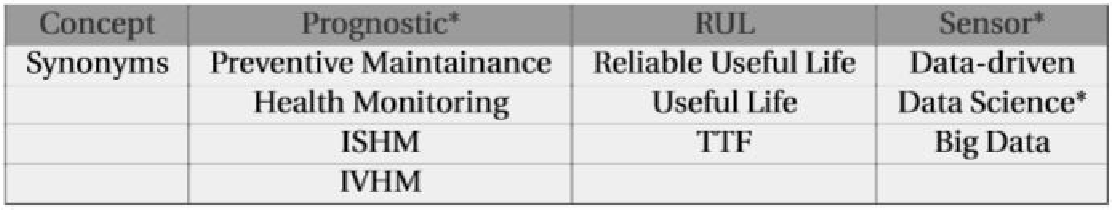
A.2. Introduction

The thesis assignment is based on developing prognostics for a Boeing 787 system regarding “Remaining Useful Life” (RUL) predictions. New aircraft generations provide more and more data, which can be analysed. There are many problems in using this data for predictive purposes. Main issues are an absent of a large event data set, many different data sources, the reliability of algorithms and translation into practical and useful support tools. A large European project can supply data of the Boeing 787 with regards of maintenance.

This assignment can help airlines and MRO organization in determining a different type of maintenance plan and improvements in the flights scheduling. The current maintenance strategy of the parts researched allows the parts to fail and replaced when needed. When the expected life of certain components can be monitored, an improvement can be made in terms of efficiency, cost and risk. Aircraft have reduced time, planning and safety can be improved. This project consists of several different steps. First explore the current state of the art in terms of data science, fault detection, diagnostics and prognostics. Secondly cleaning, integrating and processing the data. Next an algorithm is required which predicts the remaining useful life. Finally, the algorithm needs to be verified and validated.

A.3. State-of-the-art/Literature Review

Table (2.1) Concepts and synonyms with regards to the research topic



Concept	Prognostic*	RUL	Sensor*
Synonyms	Preventive Maintainance	Reliable Useful Life	Data-driven
	Health Monitoring	Useful Life	Data Science*
	ISHM	TTF	Big Data
	IVHM		

SEARCH PLAN

The search plan structure below is based on the information provided in the information literacy course 3 of the TU Delft [9]. To be able to find useful information about the research subject a dedicated search plan is required. The research subject is already given by the assignment as "Developing prognostics for B787 system remaining useful life (RUL) predictions". First the research topic needs to be divided several concepts. Secondly a search query needs to be made to find relevant articles. Next a mind map is made of the research subject to create a clearer overview of the topic. Finally, the statistics the search query are presented with the most important articles, institutions, etc.

To be able to make a search query several concepts need to be selected. First, the prognostics is the main concept of the assignment and need to be included. Secondly Remaining Use-full Life is included, since this is the required outcome of the prognostic analysis. Finally, sensor data/data-driven is included in the concepts, since on-board generated data is used in the analysis and is a big part of the assignment. An overview of the concepts and its synonyms are shown in table 2.1.

When the concepts and synonyms are selected a dedicated search query needs to be formulated. This is formulated below and is combined using OR and AND operators.

- (Prognostic* OR "Preventive Maintenance" OR "Health Monitoring*" OR "ISHM" OR "IVHM") AND ("RUL" OR "Reliable Useful Life" OR "Useful Life" OR "TTF" OR "Time to Failure") AND ("Sensor*" OR "Data-driven" OR "Data Science*" OR "Big Data")

MAINTENANCE AND PROGNOSTICS

Currently machines and systems become vastly more complex in recent years. Especially for newer transportation vehicles such as aircraft, rockets and military ships. Therefore, a completer and more integrated system is required to manage all the aspects regarding maintenance, operations and decision making.
The first company to invest in a strategy to monitor a systems status, diagnose and apply prognostics to systems and subsystems is NASA in 1990 [10]. This system was needed to directly monitor the more and more complex systems. This system is named Integrated System Health Monitoring (ISHM). This system is also sometimes referred to as Integrated Vehicle Health Management (IVHM). Prognostics is one of the main components of this health monitoring system and can really improve the overall success of the system.

Maintenance strategies are the policies on when to remove and replace components. There are in general two different global strategies. First, there is run-to-failure maintenance, where components are replaced when failure has occurred. This principle is also named reactive maintenance [11]. Secondly there is preventive maintenance, which aims to replace components before failure has occurred. These strategies are based on time, conditions or predictions made by sensor data [12].

Preventive maintenance is in current research the most interesting maintenance policy to predict RUL of a system. In industry and by academics this is divided in three global strategies.

- Interval-based maintenance: This type of maintenance management is based on determining a given interval on prior failure data. This interval can be based on the number of running-hours, flight cycles or simply on time till previous maintenance [11]. Interval-based maintenance (IBM) is widely used in maintenance scheduling and industry leading technique for maintenance prediction.
- Condition-based maintenance: Condition based maintenance uses sensor data to analyse the behaviour of a given component or system. When the system shows unacceptable behaviour maintenance actions are to be taken [13]. With condition-based maintenance health monitoring of components can be carried out before critical failure has occurred [14].
- Predictive maintenance: In predictive maintenance the RUL of the components is predicted based on features. These features can be ranging from sensor, flight data and operational profile [13]. Predictive maintenance distinguishes itself from CBM by predicting future required maintenance actions, instead of current maintenance actions. These techniques can be applied for better maintenance scheduling, since failure is predicted constantly.

For RUL prediction the most useful type of maintenance is the preventive maintenance branch. Most research in this is based on interval-based maintenance in the past and currently many condition-based maintenance strategies are following. However, predictive maintenance provides earlier warning, and this can be used to reduce cost by improving maintenance planning, reducing downtime and increasing safety. Therefore, predictive maintenance can lead to newer techniques and improved operations.

ACADEMIC RESEARCH

A set of data is required to be able to create a prognostic model. The data needs to be analysed and from this data a model is made, which indicates or predicts failure. One specific data base is referenced often by other researchers, NASA Ames prognostic data repository [15], and can be easily accessed.

The data used can be roughly qualified in three different types. First is data that is obtained from an experimental basis. Secondly there is data that is simulated using computer software and finally there is real data from a real-life application. Lei et al. [16] and Eker et al. [17] give a clear overview of the experimental and simulated data sets.

Research on prognostics has been performed in many different engineering fields. For example, in Greitzer et al. prognostics is described for a M1 Abraham tank for the US army [18]. It was applied to predict maintenance on the gas-turbine in these vehicles. In battery applications a large amount of prognostic research is performed. For battery prognostics it is important, since batteries degrade overtime. For example, Bole et al. analyses a dynamic model for battery health monitoring [19]. In this paper the 11th data set of the Data repository is used as explained by Saha et al. [20]. For engineering many papers regarding structural health of components are available. Giurgiutiu used tuned lamb wave excitation for structural health monitoring [21].

There are in general four different groups of models/algorithms in academic research to perform prognostics. The first group is knowledge-based models: a set of rules is made, where the status of the given system is predicted by these rules. These rules can be discrete rules or fuzzy rules. An example of an expert system applied for fault diagnostics can be found in the research by Simeon et al. [22]. This study showed a reduction in maintenance cost and was a successful project. However, expert systems require rules set by expert on the system, since this project only provides sensor and maintenance data results in knowledge-based systems being inapplicable.

The second model group is physical models. This model consists of an equation or several equations from empirical data. This requires a lot of research and understanding of the system to be able to develop such a complex model.

This results in this model being the most expensive and time-consuming model to predict RUL. Since no technical information is provided for this project; large amount of data is available and the given time and budget, results in physical models less desired.

One of the most common models used in preventive maintenance in industry are the statistical and stochastic models. These models mostly apply observation from the past. This is also known as empirical knowledge. The data is used without understanding of the deeper physical behaviour of the component to develop a RUL prediction. The observation is fitted on a stochastic or probabilistic model [23]. In these models data is fitted to a given model. However, in this fitting and model selection many assumptions are made. For example, in most models the data needs to be IID (Identical and Identically Distributed), all items need to be the same and not influenced by other components. Since the components on the Boeing 787 are more complex and consists of high amount of parts and uni-linear behaviour, results in this technique being difficult to apply.

Finally, a relatively new group of models is available, which are derived from Artificial Intelligence [7]. Machine learning is one of the most interesting groups in AI, since more complex and better prediction models can be obtained. In general, two different types of learning are important, supervised and unsupervised learning. Unsupervised learning is a technique to learn certain data patterns even though no required output is given. This is for example predicting the failure of a component without a failure even happening or this data is never recorded. For supervised learning, also the output of the model is given in combination with the input. The algorithm is creating a model, which links the given input to the output. The model is updated by means of optimization. For both models also a division can be made between a classification model (the predicted part is sliced in parts) and a regression model (a continuous prediction). Random Forrest is widely applied model which is robust, easy to apply and compute and can indicate the strongest features. However, the features are required to be self-engineered and selected. Also, the model is not benefiting from extreme amounts of data which are sometimes available with the current amount of sensor data. For example, Wu et al applied RF to predict tool wear [24].

An algorithm which in the recent years has gained a large amount of influence in prognostics and already applied often in data science is the Neural Network (NN) [8]. In NN's nodes are connected and the weights are changed to optimize a certain output. NN can train highly complex, large amounts of data. However, they are computational heavy, require parameter-tuning and are sensitive to overfitting. Many different types are available for RUL forecasting such as feedforward-NN, Convolutional NN (CNN) and Recurrent/Long-Short Term Memory NN (RNN/LSTM NN). The CNN and RNN/L-STM are currently state of the art and almost non-existent in industrial applications.

In terms of applying CNN for RUL prediction different approach are available. Li et al uses a deep CNN to predict the RUL of the C-MAPSS data set [25]. As an input the different features are given in a matrix multiplied with the time sequence. A CNN can select important features in the data by itself. This results in more optimal features with respect to self-engineered features.

A RNN or LSTM can optimize its network through time states. Recent research by Guo et al. used a RNN to predict the RUL of experimentally tested bearings [26] and Wu et al. applied a vanilla LSTM to predict the RUL of the NASA Ames turbofan engine data set [24]. The model showed promising results compared to other RNN models. The advantage of this of model is that behaviour of the components through their life cycle can be monitored. Early behaviour can give sings of failure in the future. Normal NN models can only evaluate a given state and do not take the past into account. However, the data needs to be complete and traceable between component replacements. If data gaps are at hand, certain problems can arise in the accuracy of the model.

The most recent thesis regarding preventive maintenance using large amount of sensor data was written by Erik IJzermans [27]. This paper also uses machine learning for predictive maintenance. This paper gives important findings in terms of improvement and other interesting findings. First, this paper uses real data to predict Flight Deck Effects (FDE) on the Boeing 747 bleed air system ten days in advance. This is extremely useful, since most other papers only rely on simple systems or on experimental/simulated data sets. On this data two types of models were used and compared. First a RF and CNN to be precise. A RNN was found to be a more suitable chose, however, due to data gaps this technique was not further explored.

From this report several recommendations are given, or improvements can be found. First applying failure data, instead of the FDE can lead to improvements, since FDE can also be caused by other errors. Applying an LSTM or CNN-LSTM is advised, since earlier behaviour of components could improve its accuracy. Another option is to change the prediction horizon of 10 days. This can be changed to another value or create a regression model. Also testing the sensitivity of the model can be tested based on quality and quantity of data. Also, a different component can be studies to see the difference in performance.

Industry application

The use of prognostics and RUL prediction in industry is mostly applying statistical data. A model, which is most commonly applied in industry is Proportional Hazard Modelling (PHM). This model uses no sensor data and only applies statistical failure data. PHM uses Time Between Failure and a baseline hazard function to predict failure based on failure observations. The advantage of this model is that no sensor data is required. Older systems do not have these sensors or they do not recorded in a data base. Also, this system provides a statistical representation of failure. Other techniques such as most machine learning techniques provide only an accuracy with no confidence interval. However, there are some applications on simpler system, such as bearings of trains [28]. In this Fumeo et al. applied a Support Vector Machine to predict failure. However, SVM can only classify linear data profiles.

So much more research can be done to implement more sophisticated machine learning techniques in industry.

Conclusion

Several research gaps that can be found in several literature types are further explained. First, applying prognostic for predictive maintenance is not used often. More focus is provided to condition-based maintenance. Secondly most research is performed using experimental or simulated data. In this research real-life data is used, which is uncommon for prognostic research on complex systems. Imbalanced data is not often considered in prognostic research and due to the data sets being simulated or experimentally set not often a problem. In recent years more deep learning techniques are applied to increase accuracy of prognostics. However, using LSTM or LSTM-CNN on real data is not applied.

A.4. RESEARCH QUESTION, AIM/OBJECTIVES AND SUB-GOALS

This section contains the research question and the research objective. These questions and objectives provide the backbone of the research and followed from the assignment and the literature research.

RESEARCH QUESTION(S)

First the main research question is provided with several sub-questions. The sub-questions are again further divided in questions. When all the questions can be answered, results in an answer to the main research question.

Research question:

Is it possible to predict the Reliable Useful Life of the Boeing CACTS system by using a Neural Network using on-board generated data?

Sub-questions:

1. Can normal operations of the CACTS system be classified?
   - (a) Which algorithm does best classify the system?
   - (b) Which failure modes are analysed?
   - (c) In what way should the data be classified?
   - (d) Which features are used to distinguish the data?
   - (e) What sensors are used in the analysis?
   - (f) Which data pre-processing methods are used?
   - (g) What is the accuracy and reliability of this classification?
   - (h) Which filtering techniques are applied?
2. What is the optimal failure horizon to predict failure in advance?
   - (a) What sort of failure horizon is preferred (Days, flights, hours or cycles)?
   - (b) What is the preferred time horizon for practical application?
   - (c) Is this optimal point fixed or variable per method?
3. Is application of CNN-LSTM network feasible for this data set?
   - (a) Does the data require the standards for an LSTM network implementation?
   - (b) How are the hyper-parameters optimized and chosen?
   - (c) What is the optimal network structure?
   - (d) How are components tracked in terms of the time horizon?
   - (e) What performance metrics are preferred?

(f) What computer and software applications are required?

(g) Can the model be applied for online data evaluation?

4. Is CNN-LSTM applicable to other systems?

   (a) Is the method applicable to other datasets?

   (b) What are the requirements for applying CNN-LSTM?

   (c) What is the robustness of the algorithm?

5. Is the accuracy and reliability of the algorithm compatible in practice?

   (a) What is the sensitivity of the algorithm with respect to quality and quantity of data?

   (b) How does the model compare to other theoretical algorithms?

RESEARCH OBJECTIVE

The research performed is practice oriented. A practical approach is required to predict failure of the CACTS system. According to Verschuren en Doorewaard [23] this is a design-oriented research. This results in the following research objective:

To achieve an accurate and reliable RUL prediction of the Boeing 787 CACTS system by means of applying a Neural Network algorithm.

This research objective is further split in several sub-goals:

1. Develop an algorithm to classify normal behaviour; -If normal behaviour can be filtered from failure impending data, results in a first step to develop a RUL prediction for the system.
2. Establish an optimal failure horizon to predict RUL; -The RUL prediction is only useful from a certain time point. For example, obtaining information of failure in more than a month is less important and less accurate.
3. Applying a CNN-LSTM network to predict the RUL; -CNN-LSTM might provide a better prediction than the two algorithms separately. The two methods already provided promising results in earlier research, both in engineering and other fields [29, 30].
4. Analysing if the CNN-LSTM model is applicable to other systems; -One of the strengths of machine learning techniques is that they can be applied without detailed knowledge of the component. This strength can hopefully be used to apply the algorithm to different systems.
5. Creating a model, which is reliable, accurate and applicable to current maintenance practice; -This is required to really make predictive maintenance more accessible. The algorithm cannot be used, when the solution is not validated properly.

A.5. THEORETICAL CONTENT/METHODOLOGY

For developing the algorithms several theories are used. Most techniques used are machine learning based. Rusell et al. provides a great overview of the complete spectrum of machine learning available [7]. In this section several classification models are explained. Another important topic is the different neural network algorithms.

To answer the first research question, there are several supervised classification models available such as hard threshold, Support Vector Machine (SVM), K-nearest neighbours and logistic regression [7, 31].

These techniques mostly divide the data in several categories, based on earlier classified data. Also, a (Convolutional) Neural Network can be applied to classify the data. These deep learning techniques are further explained by Goodfellow et al. [8].The different techniques can be applied to see what feature or method works the best and provides the best accuracy.

Also unsupervised learning can be used to detect anomalies in the data based on the sensor data itself [32].

However, these classification models are subject to unbalanced data [33]. One class of data (the failure behaviour) is present less than the majority class (normal behaviour). To overcome this, several filtering techniques can be applied and the use of the different optimization metrics. Based on the earlier methods an optimal point must be chosen to detect failure. This time horizon greatly influences the accuracy of the model and the applicability to real life applications. State of the art version of neural networks can be applied in combination for hopefully better results. A Convolutional Neural Network (CNN) uses a different architecture than only fully connected layers [34]. A fully connected layer is, where all the nodes are connected to the following layer. For a CNN a combination of convolutional layers and pooling layers are used. The convolutional layers divide the data in different features, which can be used for prediction. The pooling layers are used to reduce the amount of data and group close data points together. After each convolutional layer the data features are being more and more complex. Another useful technique is a LSTM network. It uses memory cells with gates to save a given state over multiple input steps. When the gates are closed the given memory cell cannot change it value. This results in the LSTM being able to retain information over more time-steps. In normal RNN's the information of earlier time-steps is normally lost [34]. LSTM networks are applied in many fields such as speech and handwriting recognition. The hyper-parameters of these models can be altered using a grid-search technique or Bayesian optimization [35]. This is to provide the best working model, since neural network algorithms have a vast number of variables, which influence each other. No other selection technique is available for these parameters and one of deep learning biggest challenges.

Finally, the model needs to be verified. The accuracy and reliability need to keep up with current maintenance standards. This requires accurate metric, which considers unbalanced data [36].

A.6. Experimental Set-up

In this research data is used from the ReMAP research programme. This data is not required to be gathered by myself and is available to the TU Delft. This data consists of sensor values of the Boeing 787 CACTS system. These aircraft are operated by KLM and are partner in the ReMAP programme. Analysing this large amount of data requires a large amount of computations. The data requires several steps before being able to be analysed further. First the data is required to be cleaned of errors. Next the data is required to be linked to a certain aircraft/component. This is required to connect the right sensor data to the correct failure data. Another step requires to link the failure data to a given aircraft. Next the data is required to be uncensored and made complete. Finally, a timeline can be created with the given components and their respected sensor and failure data. This timeline can be used to analyse the different components and is required for the use of the LSTM network. A continuous data flow of each component is required in chronical order to be able to track the degradation of the component over time.

This analysis will be done by using Python. Python is a programming language with good computational performance, many different machine learning algorithm packages and easy to implement with many different lectures, tutorials and support online. PyTorch for example can be used to run neural network algorithms with python. This combined with NVIDIA CUDA can compute large programs on the computers GPU and reduce the computational time significant.

A.7. Results, Outcome and Relevance

The main results of the thesis are based on the main research question: Is it possible to predict the Reliable Useful Life of the Boeing CACTS system by using a Neural Network using on-board generated data? To do so the main sub-questions help in finding this answer. The first results and outcomes would be relating the classification of normal behaviour versus failure behaviour. This is to see if the data is showing indications of following failure. Next an optimal failure horizon is required. Finally, the CNN-LSTM algorithm will hopefully result in an accurate model to predict the RUL of the CACTS model. The results can be made in steps or a regression model. The results of the accuracy of the CNN-LSTM model will be tested on the reference data and evaluated with other techniques. The validation and verification consider the aspects of unbalanced data, overfitting and robustness of the model. This can for example be done by using metrics based the confusion matrix or other strategies.

The relevance of this research project can be for academic research and industry. For academic research very few researches are done on real-life engine data. Especially the use of CNN-LSTM network on real-data is non-existent. Also finding gaps in the use of real-life data with machine learning techniques can help academics to improve further research. On an industry level, this research provides one of the first researches based on the CACTS system and will show if prognostic research is possible on the system. This thesis might provide a robust and useful system to predict the RUL. When implemented this leads to reduced cost, improved maintenance scheduling and increased safety.

A.8. Project Planning and Gannt Chart

An overview of the general project planning can be seen in appendix A. The planning is divided in three parts: Literature study, pre-mid-term and post-mid-term. The literature study is finished with a literature review document. The next phase is started with a kick-off meeting at 25/11/2019. In the following phase the model is developed in several steps. First the data requires alterations and modifications. First analysis is performed on the PHM08 data set, since at the start of the project the ReMAP data is not available. Then the anomaly detection, optimum failure horizon and CNN-LSTM algorithm needs to be developed. This also includes three weeks of holiday. More data analysis is required when the ReMAP data is available. This is concluded with a mid-term meeting at the beginning of March.

Next the model requires sensitivity analysis, verification and validation. Next the report needs to be finalised. After a draft review the green light review is done. Finally, the thesis in handed-in and the final thesis defence is planned around 29/06/2020.

The Gannt chart is, however, an overview of the major tasks at a current time stage. At some point in time iteration might be required to improve earlier steps of the research.

A.9. Conclusions

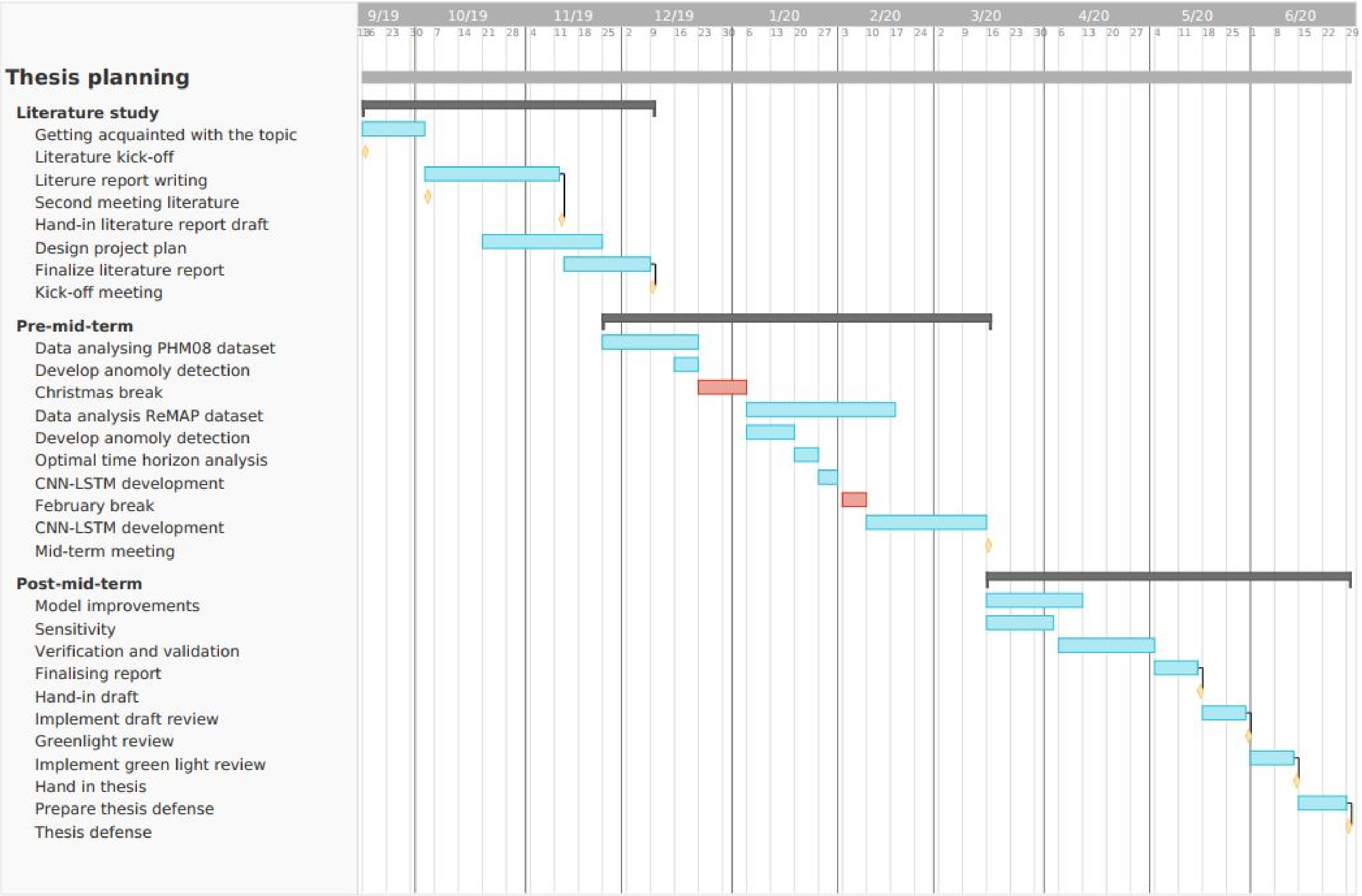
The main thesis assignment is to develop prognostics for a Boeing 787 system regarding RUL prediction. These predictions can help provide better, cheaper and safer maintenance on these aircraft. However, the main problems in aircraft maintenance is small event data sets, many different data sources, a reliable and effective algorithm and translation into a practical tool. From the literature study a trend towards predictive maintenance was shown. In predictive maintenance sensor data is used to predict the RUL of components. A lot of research has been performed by academia on this topic, however most research is performed with experimental data. Also, most systems that are researched are mostly less complex than the Boeing 787 systems. Finally, more complex deep learning techniques are used to predict the RUL op components. In earlier research a CNN-LSTM algorithm has many advantages and provides an interesting approach.

This resulted in the research question: Is it possible to predict the Reliable Useful Life of the Boeing CACTS system by using a Neural Network using on-board generated data? With several subquestions and objectives that lead to answering of this question.

The methodology used is based on machine learning techniques. First the data is tested with several supervised classification models to detect anomalies in the data and predict incoming failures. Next an optimum point for RUL prediction is required to be found. Next the CNN-LSTM algorithm is required to be developed with finally, a sensitivity analysis, verification and validation.

The data is provided by the ReMAP research programme. However, this data does require cleaning and analysis to link all the factors. The final analysis will be done using Python. This has advantages in terms of computational time, packages and implementation.

Finally, a Gannt chart is provided with all the major milestones and tasks. This chart gives an overview of the complete project and is the main guideline of the project execution.


9/19
10/19
11/19
12/19
1/20
2/20
3/20
4/20
5/20
6/20
Thesis planning
Literature study
Getting acquainted with the topic
Literature kick-off
Literure report writing
Second meeting literature
Hand-in literature report draft
Design project plan
Finalize literature report
Kick-off meeting
Pre-mid-term
Data analysing PHM08 dataset
Develop anomoly detection
Christmas break
Data analysis ReMAP dataset
Develop anomoly detection
Optimal time horizon analysis
CNN-LSTM development
February break
CNN-LSTM development
Mid-term meeting
Post-mid-term
Model improvements
Sensitivity
Verification and validation
Finalising report
Hand-in draft
Implement draft review
Greenlight review
Implement green light review
Hand in thesis
Prepare thesis defense
Thesis defense

B
Literature study

Previously graded under under AE4020

B.1. INTRODUCTION

New aircraft have more and more data generated on-board. This data can be used for many different operational advantages. One of these advantages is using this data to predict failures. Leading to increased up-time, better scheduled maintenance and increased safety. However implementation of such a model in the aircraft industry has a number of challenges. The number of failure occurrences on aircraft is extremely low, resulting in a small event data set. Sensor measurements and failure data needs to be combined in a model, which can accurately predict failure behaviour. The accuracy needs to be precise and reliability of the model needs to be high to be applicable. Finally the model also needs to be practical and preferably applicable to more than one system. A major European research project named ReMAP provides data to perform maintenance research on selected Boeing 787 parts. This results in the following research objective:

Developing prognostics for B787 systems regarding RUL predictions.

This literature study is performed before the start of the master thesis to become more acquainted with maintenance, prognostics and RUL prediction get up to date with the state-of-the-art in RUL prediction. However, the main goal of the literature study is to find gaps in the current literature and formulate a research question. This literature study is divided in a number of sub-questions.

- Which maintenance strategies are available and which is most suited for RUL prediction?
- What are the major applications and important factors for RUL prediction?
- Which models can be used for RUL prediction?

The report is structured in the following way. First a small chapter is given with global information about the search method. This shows the general concepts and statistics about the research topic. Afterwards the different maintenance management techniques are explained, with a focus on the different preventive maintenance techniques. In Chapter 4 the different applications and data sources are explained. Examples are given of different applications and further information is provided about health indicators, health stages and imbalanced data. In chapter 5 an overview is given of the different models used in research and industry. Next a review of an earlier performed thesis is performed and finally a conclusion of the literature study with the proposed research question.

B.2. SEARCH METHOD

To be able to obtain and find suitable literature, a dedicated search method is required. In this chapter an overview of the literature sources is given. Next a search plan is provided to find useful information. Finally, the literature processing is described.

LITERATURE DATA

In this literature review, data is used from various scientific resources as mentioned by the TU Delft library [37]. These article data bases allow researchers to simply find already existing articles, papers, journals and conference papers about a certain topic. These data bases also provide statistics and can easily filter on author, year of publications and relevance [37]. To be able to find the best information a search plan needs to be made as described in Section B.2.

SEARCH PLAN

The search plan structure below is based on the information provided in the information literacy course 3 of the TU Delft [9]. To be able to find useful information about the research subject a dedicated search plan is required. The research subject is already given by the assignment as "Developing prognostics for B787 system remaining useful life (RUL) predictions". First the research topic needs to be divided a number of concepts. Secondly a search query needs to made to find relevant articles. Next a mind map is made of the research subject to create a clearer overview of the topic. Finally the statistics the search query are presented with the most important articles, institutions, etc.

Concepts To be able to make a search query a number of concepts need to be selected. First, the prognostics is the main concept of the assignment and need to be included. Secondly Remaining Use-full Life is included, since this is the required outcome of the prognostic analysis. Finally, sensor data/data-driven is included in the concepts, since on-board generated data is used in the analysis and is a big part of the assignment. An overview of the concepts and its synonyms are shown in Table B.1.

Table B.1: Concepts and synonyms with regards to the research topic

Concept	Prognostic*	RUL	Sensor*
Synonyms	Preventive Maintainance	Reliable Useful Life	Data-driven
	Health Monitoring ISHM IVHM	Useful Life TTF	Data Science* Big Data

Search Query When the concepts and synonyms are selected a dedicated search query needs to be formulated. This is formulated below and is combined using OR and AND operators. This search query is further used to evaluate a mind map and provide article and research statistics on this topic.

- *(Prognostic* OR "Preventive Maintainance" OR "Health Monitoring*" OR "ISHM" OR "IVHM") AND ("RUL" OR "Relaible Useful Life" OR "Useful Life" OR "TTF" OR "Time to Failure") AND ("Sensor*" OR "Data-driven" OR "Data Science*" OR "Big Data")*

Mind map From the information gathered during orientation and the articles found a global mind map is made with some of the major topics. This includes the applications. Where is prognostics used and what are its benefits. Secondly on what parts is it used. Next there is a large number of different models and algorithms, which can solve the problem. Finally, the data that is put in the system, needs to be modified with several techniques to be able to be used in the algorithms. This overview is just the surface of the topic. More in depth information about these topics and many other are further specified in the rest of the report.

Statistics To be able to analyse the relevance of a certain topic, a statistical overview can be given. An overview of the number of published articles, most cited author, institution with the most publications and the general subject area are given in figure B.1 - B.3.

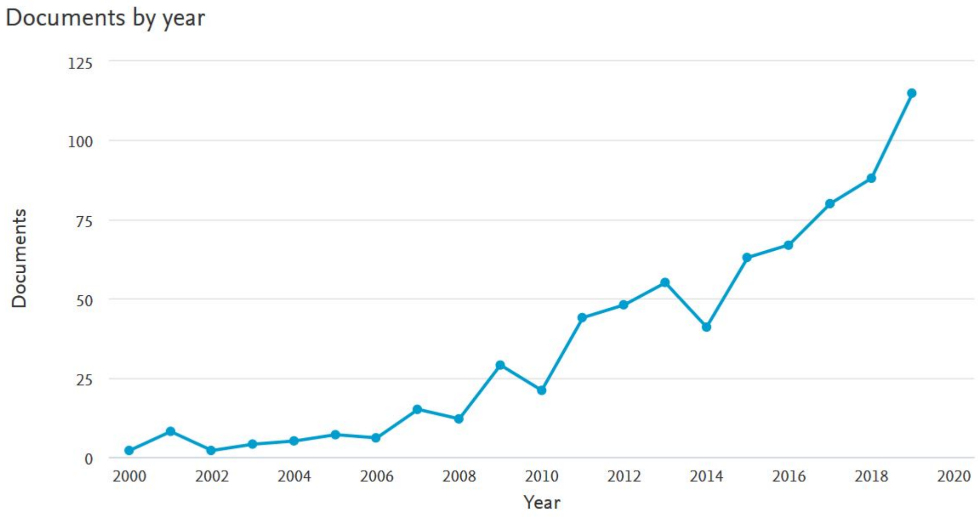


Figure B.1: Published documents by year on Scopus with the search query in section B.2 from 2000 till 2019.

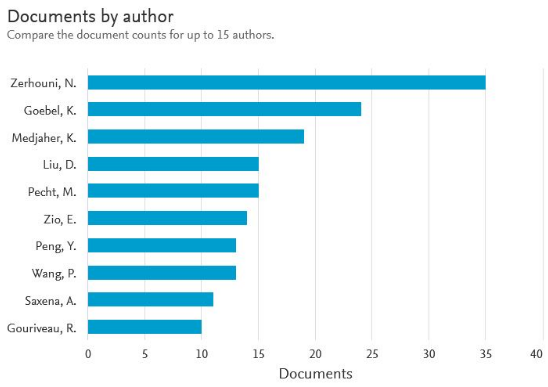


Figure B.2: Number of document per author on Scopus with the search query in section B.2 .

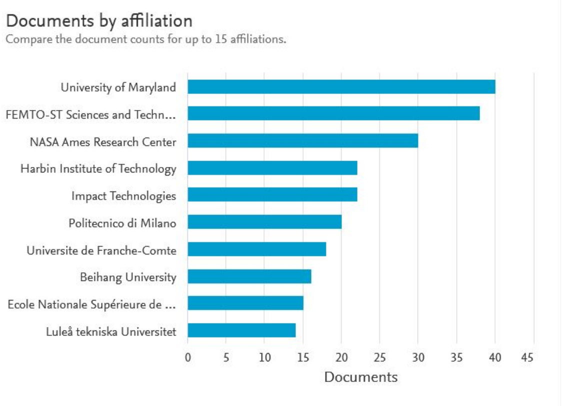


Figure B.3: The number of document on Scopus with the search query in section B.2.

B.3. MAINTENANCE MANAGEMENT

In this chapter different maintenance methods are specified and the origin of the newer maintenance techniques. This information gives the reader a broader view of the different topics and helps in the understanding and finding of new literature papers on this topic. In terms of maintenance strategies, a large amount of different terminology and frameworks are used. This report shows not all the different terminologies and frameworks but gives a framework, which works best with regards to the research objective and the given topic.

ISHM

Currently machines and systems become vastly more complex in recent years. Especially for newer transportation vehicles such as aircraft, rockets and military ships. Therefore, a more complete and integrated system is required to manage all the aspects regarding maintenance, operations and decision making.
The first company to invest in a strategy to monitor a systems status, diagnose and apply prognostics to systems and subsystems is NASA in 1990 [10]. This system was needed to directly monitor the more and more complex systems. This system is named Integrated System Health Monitoring (ISHM). This system is also sometimes referred to as Integrated Vehicle Health Management (IVHM). Prognostics is one of the main components of this health monitoring system and can really improve the overall success of the system.

The overall system ISHM results in the following benefits as described by NASA. First of all, the safety of the system is increased, which therefore, increases the success of the mission. Secondly, improving different operations aspects such as, increasing operation time, reducing labour hours and reducing costs. This will lead to higher up-time of the aircraft.

Maintenance Strategies

Maintenance strategies are the policies on when to remove and replace components. There are in general two different global strategies. First of all, there is run-to-failure maintenance, where components are replaced when failure has occurred. This principle is also named reactive maintenance [11]. Secondly, there is preventive maintenance, which aims to replace components before failure has occurred. These strategies are based on time, conditions or predictions made by sensor data [12].

To some researchers preventive maintenance is equivalent to time-based maintenance and predictive maintenance is equal to condition-based maintenance [3]. When reading a report keep this in mind and quickly observe which terminology the researcher is using. Another observation is that predictive and condition-based maintenance is considered the same for some researchers [11]. In this research reactive and preventive maintenance are split. Preventive maintenance is further divided in interval-based, condition-based and predictive maintenance.

Reactive maintenance

This management type allows components fail. Afterwards maintenance is being applied to restore the system. This type of maintenance is the easiest and oldest type. No maintenance will be provided before the actual failure has happened [3]. There are several strengths to this maintenance strategy. first of all, the part are only replaced when they break down. This results in the parts being used to their full potential and life-span. Secondly, no data base is required with all the failure data that has occurred nor sensor data of the system.

However, in terms of maintenance this system is difficult and costly technique. This type of maintenance results in high downtime, high labor cost in terms of overtime and a large stock is required [3]. Especially in aviation maintenance this type of maintenance can cause problems, since the aircraft can be located in many different locations when maintenance is required. On all the locations a high amount of stock and maintenance personnel is required.

Preventive maintenance

Another strategy which aims to reduce maintenance cost and increase the overall up-time and reliability of systems [38]. These maintenance tasks consists of several

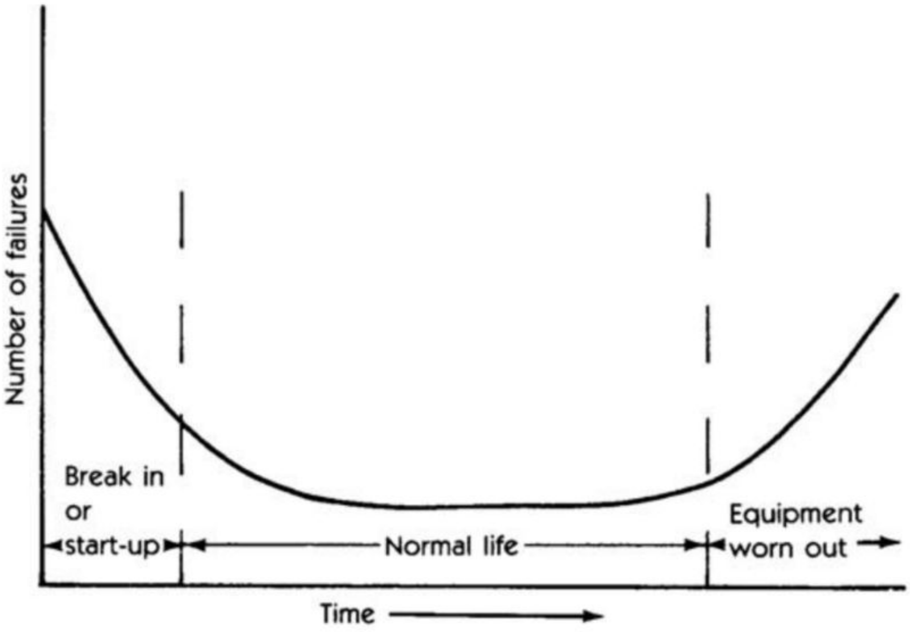


Figure B.4: Illustration of the three life-phases in IBM [3]

different actions such as lubrication, inspections and adjustments to the components. There are globally given three methods based on interval, the condition of the component and finally, prediction based on sensor and behaviour.

Interval-based maintenance This type of maintenance management is based on determining a given interval on prior failure data. This interval can be based on the amount of running-hours, flight cycles or simply on time till previous maintenance [11]. Interval-based maintenance (IBM) is widely used in maintenance scheduling. IBM assumes that the failures of a system are predictable and are based on the so-called bathtub curve. The failure data of the system is analyzed and plotted. For the bathtub curve, three regions can be distinguished. Firstly the burn-in failures. In this region failures are mostly more common, since new components can be afflicted with manufacturing errors. Next the useful-life phase. In this phase the failure rate stays mostly constant and replacement at this point is not needed. Finally, the end-of-life failure phase is present. This results in an increase in failure rate and causes more failure events. An overview of these phases can be seen in Figure B.4. The most common distribution to fit the failure data is the Weibull distribution [39].

IBM has several strengths in comparison to RM. Firstly an analysis can be made to when to replace a given component. When the component is in its wear-out phase, the replacement of the given part can become of more interest, since RM is more costly. Secondly, the data can be also be used to set-up a periodic maintenance schedule. With this schedule a large amount of abrupt failures can be reduced, since parts are replaced before failures even occurs.

However, IBM has many assumptions and requirements, which all imply their own weaknesses. first of all, a large amount of failure information is required. This is required to develop the failure rate curve and provide reliable predictions. When too few information is available, the given distribution can be weakly fitted to the data. Especially with aircraft data this a problem, since aircraft components are generally made to fail as little as possible and results in low amount of failure events. Secondly, parts are subject to imperfect repair. This is when a given component is not restored to its original state. However, there are strategies and models to take imperfect repair into account [40]. Another aspect regarding the bathtub curve is that it is only applicable to a small amount of components [41]. More than 80 percent of the components do not show any end-of-life reliability phase. This makes this technique only applicable to a hand full of components and not to the majority of the systems. Finally, the assumption that all components are identically distributed, thus have the same failure distribution. This ,however, is most of the times not the case, since all components are slightly different due to manufacturing or maintenance differences.

Condition-based maintenance Condition-based maintenance uses sensor data to analyse the behaviour of a given component or system. When the system shows unacceptable behaviour maintenance actions are to be taken [13]. With condition-based maintenance health monitoring of components can be carried out before critical failure has occurred [14]. Maintenance work can be initiated before the component has failed. With CBM a system can be monitored based on the direct condition of the component and prevent unnecessary maintenance and part can be replaced in time. However, no maintenance suggestions are only given when anomalies are showing.

This will result in the improving of maintenance in two ways. Unnecessary maintenance is prevented and the reliability is improved, since parts can be monitored and acted accordingly. However, large amount sensor and data solutions are required to monitor failure. This also requires a good algorithm to find anomalies in the data using supervised or unsupervised techniques.

Predictive maintenance The final group of preventive maintenance techniques is predictive maintenance. PM analyses the condition of the components and predicts when maintenance will be required. This is one step more advanced than condition-based maintenance, since current conditions are used to predict future failures. In predictive maintenance the RUL of the components is predicted based on features. These features can be ranging from sensor, flight data and operational profile [13]. These techniques can be applied for better maintenance scheduling, since failure is predicted constantly. The predictive maintenance policies are for this research the most interesting, since a RUL needed to be provided. However, many condition-based strategies can be analysed to detect if there are anomalies in the first place.

Prognostics

First the definition of prognostics needs to be clear and concise. In Sikorska et al. [23] different definitions of prognostics are summarized. These definitions represent four different statements. First prognostics should be applied to a component or at sub-component level. Secondly, prognostics aims to predict the life time of a component when a certain failure is occurring until its failure. Thirdly, an overview of the component use in the future is required and finally is stated that "prognostics is related to,but not the same as, diagnostics". The fitting description of prognostic is according to ISO13381-1: ’an estimation of time to failure and risk for one or more existing and future failure modes’ [42]. Close related to prognostics is the estimation of Remaining Use-full Life (RUL). This remaining useful prediction shall have the following criteria [12]. The system/component is certain to be able to operate until the end of life. Secondly, the system can contain a large amount of different failure modes. All of these different failure modes shall have their own RUL. Finally, the RUL values have to be have a certain uncertainty level. This is to better indicate the precision and reliability of the method.
Many researchers expand the principle of prognostics towards a more complete system, named Prognostics and Health Management (PHM) [12, 43, 44]. PHM is a larger system which combines the complete system to includes monitoring, fault detection, tasks etc. and reduces with this the total cost and increases safety [45]. This is very close related to ISHM. However, PHM uses prognostics as its basis, while ISHM can also be done using condition-based maintenance and can include prognostics.

Conclusion

The solution for predicting failure of components is being researched for already 30 years. Many of these techniques require different types of data, checks or knowledge of the given components. For RUL prediction the most useful type of maintenance is the preventive maintenance branch. Most research in this is based on interval based maintenance in the past and currently many condition-based maintenance strategies are following. However, predictive maintenance provides earlier warning and this can be used to reduce cost by improving maintenance planning, reducing downtime and increasing safety. Therefore predictive maintenance can lead to newer techniques and improved operations.

B.4. Applications and data sources

Applications

There are many fields where prognostics are applicable other than engineering and are already used for a longer period of time. For example, prognostics were already used in other industries such as medicine [46] and weather forecasting [47]. In medicine prognostics is used to provide patients and doctors information of the progress of a disease and give the patient a survival percentage and time. For example, in [48] a neural network is used to distinct four different growth patterns of cancer. In terms of weather forecasting it is very useful, since this is the main driver for current weather forecast.

Next a number of examples in terms of engineering are given and its strengths and weaknesses are summarized. This will include different engineering disciplines and aerospace examples.

Military

Prognostics were already used for military applications for a long time. For example, in Greitzer et al. [18] prognostics is described for a M1 Abraham tank for the US army. It was applied to predict maintenance on the gas-turbine in these vehicles. The methods used however are in terms of prognostics reasonable old. Most military applications are not open research papers and hard to obtain.

Battery

In battery applications a large amount of prognostic research is performed. For battery prognostics it is important, since batteries degrade overtime. For example, Bole et al. [19] analyses a dynamic model for battery health monitoring. In this paper the 11th data set of the Data repository is used as explained in B.4. Saha et al. uses particle filtering technique to estimate the degradation of battery capacity with the 5th set provided by NASA Ames. These techniques can be analysed to help structure component failure, however battery degradation is always present and shows a clear degradation pattern. For predicting the RUL of engineering components this degradation pattern might be harder to find.

Structural

For engineering a large number of papers regarding structural health of components are available. Giurgiutiu [21] used tuned lamb wave excitation for structural health monitoring. This research was already carried out in 2005 and therefore, a simpler method is used. Sbarufatti et al. [49] uses a combination of different methods to calculate structural failure. It uses sequential Monte-Carlo sampling to predict structural failure of the crack, while the crack growth itself is simulated using a neural network.

The prognostics for structural components however, often use the relation of the size of a crack or other impurities in the material to predict an impending failure. This is not particularly useful when analyzing different component, which do not show a clear and traceable damage sign. Also SHM can be mostly used for a single phase or type of action. However, SHM is currently not used in real flying aircraft structures during normal operations.

Types of Data sets

A set of data is required to be able to create a prognostic model. The data needs to be analyzed and from this data a model is made, which indicates or predicts failure. One particular data base is referenced often by other researchers, NASA Ames prognostic data repository [15], and can be easily accessed. It currently consists of sixteen different data-sets, which are all aimed to help researchers create and test new prognostic models. The data used can be roughly qualified in three different types. First is data that is obtained from an experimental basis. Secondly, there is data that is simulated using computer software and finally there is real data from a real life application. In the following sections, examples and strengths and weaknesses of the different data-set types are given. Lei et al. [16] and Eker et al. [17] give a clear overview of the experimental and simulated data sets.

Experimental set-up Data is obtained by means of testing a set of components in a controlled environment with this strategy. This technique is widely used and many research is based on these sets [16, 20, 21, 49]. It is mostly applied to bearings, structural components and batteries. In a controlled environment a test is conducted. This can sometimes be done under extremer conditions to accelerate the degradation process. This is to be able to decrease the time per sample to show failure. In this experimental set-up it is easy to measure, analyze and control the test.

The strength of this method is that the sensor required can easily be installed. For example, in the bearings test vibrations sensors are widely used. If a dedicated sensor was required to be installed in an operating system, would result in a more difficult implementation challenge, due to safety and costs. The data can also, be obtained directly, which ensures that data gaps and losses are prevented. Finally, the direct time of failure is known by means of observation. This gives no censoring in the data. However, this type of test is not an appropriate representation of normal operations. The data is not subjected to the environmental changes in which the normal operation is present. This can make a large difference, especially when sensors, such as temperature and pressure are used. Most of these experimental set-ups use a single component for testing. However, in normal operations failures will occur due to interactions in the complete system of components.

Simulated data Another type of data, which is commonly used, is related to computer simulated data [16, 25]. This data consist of run to failure data of a given system, which is used for prognostic purposes. One of the mostly used data sets is the Turbo Engine Degradation Simulation Data Set [15]. This set uses C-MAPSS data to simulate four different sets of data with different operational conditions. The failure data of these engines is also given. This model consists of 14 inputs and 58 outputs, which result in useful data to predict certain components in a turbofan engine [25]. Of all the 58 outputs, 21 are used for prognostics. In 2008 this set was used to develop prognostics in the IEEE PHM 2008 conference [50].

The strengths of this type of data is that it represents the complete system. This results in failures, which can occur due to a combination of components. Also the model can create a very large amount of data, which can be used in highly advanced algorithms.
However, this type of data model has several drawbacks. The connection between reality and this model can create problems when applying it to real engine data. The real turbofan engines provide only very little run-to-failure data. So a very data hungry machine learning technique less accurate or even not possible in the real world. Some researchers even used it to large set of training data to find unit-to-unit similarities in the data [51, 52]. Finally the data set is simulated as close the original degradation of the engine, however, it still does not capture all the different processes.

Real life application Condition-based or predictive maintenance is currently not applied in many real applications and is difficult to find in literature. Most of the research is done on controlled data packages. There are several different articles, which do use real life data. However, the components researched are mostly not complex (e.g. bearings). For example, for railway applications. Fumeo et al. [28] uses condition-based maintenance to predict the RUL of train axle bearings to improve operational efficiency. This was applied in a real case study, where data was streamed from the trains. This case study showed that the proposed methodology brought a benefit to the current operations.

Previous theses of aerospace engineering master students at Air Transport and Operations did use real data as the basis for their methods. Prins [53] uses a neural network for prognostic purposes using real engine data. The most recent paper regarding preventive maintenance using large amount of sensor data was written by Erik IJzermans [27]. This paper also uses machine learning for predictive maintenance. Further explanation on the different machine learning models are available in Section B.5. This paper gives important findings in terms of improvement and other interesting findings. It is a combination of own findings and the recommendations in the report.

First of all, this paper uses real data to predict Flight Deck Effects (FDE) on the Boeing 747 bleed air system ten days in advance. This is extremely useful, since most other papers only rely on simple systems or on experimental/simulated data sets. On this data two types of models were used and compared. First a RF and CNN to be precise. A RNN was found to be a more suitable chose, however, due to data gaps this technique was not further explored. The data was pre-processed with several steps such as linking FDE data, uncensoring data and time line recreations. The hyper parameters were found for both RF and RNN with a grid search technique. The main metric used were the Average Precision and ROC curve.

From this paper many different interesting recommendations were given. First of all, applying a LSTM NN or RNN to improve the performance would be a great addition. However, the data for the Boeing 787 in this research needs to allow this. Also using a CNN-LSTM structure is promising. It can leverage the potentials of finding location invariant features, combined with finding features over time. Another suggestion is to add engineered features to the input of a NN, combined with the raw input. This might improve the accuracy of the model. Another suggestion is to combine a deep learning method with a survival model or a proportional hazard model (Section B.5). Further more increasing the amount of sensitivity analysis by decreasing the amount of inputs or adding errors to see the change in accuracy.

Finally two important improvements can be done. First of all, the model only divides the data in healthy and 10 days till FDE. This can be altered in two different ways. It can be divided in different steps till failure or prediction horizon can be optimized. This results in the model being a classification algorithm. Another approach is to make the model a regression model and let the RUL prediction become smaller till failure. Secondly, the model uses FDE as failure prediction. However, FDE's to do not always imply failure as mentioned by IJzermans. There might be particles blocking the FDE sensor or computer errors. Validating the FDE errors with shop reports is a very tedious task to remove these errors. This results in the model learning good behaviour as wrong in some cases. However, for the Boeing 787 system there are a multitude of different failure documents which can be used, besides the FDE's. This does improve the model accuracy.
The recommendations given here should be taken into account when writing the thesis. They provide valuable information and help in further research applying real data. Since there are few reports available which use real life application data, leads to a science gap. More case studies with real data are required to be performed in general. By researching with real results, gives a better understanding of the real world. This can result in new findings/problems, validation and even creating a business case, where the gains of a prognostic system can be analysed. Another gap is that most system are reasonably simple. Complex systems are not that often covered in literature and in most research the given part is already thoroughly analysed. There for analysing a more complex system can be of interest for a new research.

HEALTH INDICATORS HEALTH STAGE

Several steps need to be taken when analysing data. For processing data for prognostics two different principles are important. The health stage indicates the condition of component and the health indicator is which feature is analysed. In this section these two concepts will be further explained.

Health stage Components in general can show a large variety of degradation behaviour. A component can show degradation behaviour from the beginning of its lifetime or only after a given time/cycles. There are even possibilities of more stages, for example, in the double row bearing data set of NASA AMES, three different stages can be found [54]. First there is a healthy stage in which the component does not show any signs of failure. Secondly, a section can be found in the data, which shows degradation behaviour and finally a region can be found, which shows unhealthy behaviour. These different part of data, which show different failure and degradation characteristics.

The division in health stages is often used in different models to predict in which health stage the component is operating. These models are named classification models and predict the RUL of a component. The classification model is used in prognostic research in the majority of the articles [27, 55**?**]. This classification model determines when a component is behaving in a way where failures are highly applicable. This is normally done by setting a threshold on when maintenance can be done and performed in advance. Giving advice on failure of a component that will fail in the following hour is not as useful as for example, 10 days.
Another method to determine the RUL is by having a continuous regression model. This model is not divided in number regions but gives a non-discrete representation of when failure will occur. A combination of a classification model and a regression model is also possible. Further explanation on regression and classification model can be read in Section B.5.

Health Indicators Health indicators are used to evaluate the state of the component or establish a RUL. It can be classically done by inspection, however, this is in some cases not possible or very expensive. For example, it can not be possible due to failures at micro level or simply not easy to reach at a frequent interval [16].
Another approach is to monitor a number of parameters of the system, to analyse its behaviour directly. However, a given parameter does not give you any information about the state of the system. These parameters need to be analysed to be able to monitor the system.

This monitoring can be done in many different ways. First of all, a single value can be analysed and if this exceeds or is lower than a certain value, the system is about to fail. This is also known as a Physical Health Indicator [16]. This however, is commonly not given by a single parameters for more complex systems. These values are combined, modified or used to create statistical values (e.g. skewness, kurtonis and averages) or a combination of these. These features hold valuable information on a condition of a part. The data however, is losing its physical meaning and is therefore also named as a Virtual Health Indicator.
These features can be chosen in an almost infinite amount of combinations. This

makes it very hard to select the right features and come with the best solution. For this research it might be interesting to look through the data to find relevant features or failure signs.

For combining a given number of features, a method named First Principle Component can be used. This method transforms the data in in a different, new coordinate system. In this new coordinate system the greatest variance is projected on the first coordinate, the second highest is projected on the second coordinate and so on. It reduces the number of dimensions in essence [32]. This is can be used for example in unsupervised learning, which will be explained in Section B.5. Other methods to reduce the dimensionality of data is for example, prescribed by Fordellone et al. [56].

If the failure data is known, different approaches can be applied to select the right features. The feature selection in these methods is mostly linked to the model used and its efficiency. Most research is performed in terms of classification models or regression models. A number of supervised-learning methods can be used to give the feature, which is the most leading in the model. For example, a Random Forest can be used to quantify the most important features [27].
Another approach to create features is by using a Convolution Neural Network [25]. CNN is a type of machine learning, which is able to capture important dependencies. This results in that a CNN can distinguish a set of features without pre-processing them. The network selects the critical data itself after enough training. A CNN can in this way predict a RUL on features we normally don't consider [8]. This technique is already commonly used in terms speech and picture recognition, but only in recent years to other applications such as prognostics.

Imbalanced Data

Imbalanced data is a problem in machine learning, where a certain type of behaviour or class is present far less often than another class. In terms of aircraft maintenance this is highly applicable, since close to failure behaviour is less present than normal behaviour. For imbalanced data where the majority class is present in the order of 100:1 (or even higher) a problem in classification occurs. The majority class is predicted almost correct, whilst the minority class is only predicted correctly around 10 percent. This is a big problem, since false classifying the minority class can lead to catastrophic failure. A well structured overview of imbalanced data and techniques are given by He and Garcia [36].

The majority class is classified better, since the classification algorithms mostly divide their data in sets of groups as described in Section B.5. For example, a RF will create more trees regarding the features of the majority class and less of the minority class. In this procedure the already scarce data will be even less evaluated.

The problems and solutions regarding this technique will be described in this section. By using different data techniques this problem can become less important. Current solutions for imbalanced data are: sampling methods and cost-sensitive methods. Another solution for handling imbalanced data is the use of advanced error metrics [36]. These advanced metrics optimize on different parameters, than for example, the overall accuracy.

Imbalanced methods In this section the different methods to process imbalanced data are prescribed.
Sampling methods:
The main idea of this technique is to modify the data that consists of a balanced distribution. The first type is random under- and oversampling. With these techniques the data is made equally distributed. In undersampling, data points of the majority class are removed randomly and for oversampling the minority class is randomly duplicated. With this technique it is possible to increase the accuracy of classifying the correct type. However, both models have their own weaknesses. Oversampling can result in over fitting of the data, since the features of training data simply duplicated. Undersampling can result in the majority class being classified less accurate, since important features might be removed. Informed undersampling reduces this weakness, especially the BalanceCascade algorithms [33].

An advanced oversampling technique named Synthetic minority oversampling technique (SMOTE) has shown great performance [57]. With this technique the minority data points are added using a K-nearest neighbour Algorithm (as described in Figure B.5). However, this technique can have problems in terms of variance and generalizations. This disadvantage has been overcome by Borderline-SMOTE and Adaptive Synthetic Sampling (ADA-SYN).
Finally sampling techniques based on clustering, using k-means clustering and combination of different techniques such as SMOTE and boosting [58].

Cost-sensitive methods: This method is not trying to compensate the minor class in terms distribution with the majority class, but gives different "cost" values in making a certain error. Making a failure for determining a minority class falsely is more costly than making an error in the majority class. This type of method can be applied to many different algorithms. It can be applied to simple machine learning techniques (SVM, Bayesian classifier), but also to Decision Tree techniques such as RF or NN techniques. The NN uses cost-sensitive learning to alter the back-propagation technique to better predict imbalanced data [59].

These techniques all provide valid methods for better accuracy in terms of classification. However, these methods are providing different result for different applications. Therefore, as recommendation testing of suitable sampling method, such as ADA-SYN, and a cost-sensitive method, based on the type of algorithm used, would be useful. Afterwards the best method can be used and implemented.

Imbalanced metric A metric is required for analysing the result of a given algorithm. For imbalanced data the following metrics can be used: singular metrics, ROC curves, PR curves and cost curves.

Singular metrics:
This type of metrics uses a formula using data from the confusion matrix. A confusion matrix is divided in True Positives and True Negatives. These values are predicted by the model to be in the correct class. On the other hand, False Positives and False Negatives are predicted wrong. In Equation B.1 the formula for accuracy is shown. This metric is one of the most frequently used. It simply gives a percentage of the correct predicted values. When the data is highly imbalanced this causes a problem. If the model simply labels everything to be the majority class a high accuracy is reached, whilst the model is not useful.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN)} \tag{B.1}$$

Further examples of metrics are Precision and Recall (Equation B.2 B.3). Precision is testing if the positive class predicted correctly and recall is measuring the percent of positive measurement where selected correctly (completeness).

$$\text{Precision } = \frac{TP}{TP + FP} \tag{B.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{B.3}$$

ROC and PR curves:

In this metric the True Positive rate is plotted against False Positive rate. In Equation B.4 the formulas are provided. With this technique a line can be drawn based on a different models [60]. For prediction models that provide continuous output, such as a regression model, a complete line can be drawn. When the lines are given, it can be used for choosing and optimizing the right technique. If a data point is accurate, random or worse than random.

Another metric that can be used are Precision-Recall curves [61]. These are based on the earlier mentioned Equation B.2 and B.3. This metric is useful for imbalanced data, especially when skewed data is applicable. ROC curves can have problems when dealing with imbalanced data.

$$TP_{-rate} = \frac{TP}{TP+FN}; \quad FP_{-rate} = \frac{FP}{FP+TN} \tag{B.4}$$

Cost curves:

A different variant for ROC and PR curves are cost curves. In this visual representation of the cost of wrongly classifying the positive class [62]. With this technique FN and FP can be given different costs to truly choose the optimum technique. For example, in engineering, false prediction of a failure (FP) is far more costly than false classifying no failure (FN).

Concluding:

In conclusion for the use of metrics for imbalanced data. The use of PR curves and cost curves is most valuable. They provide a useful overview and can be used to compare different algorithms. Accuracy and error rate can however not be used for unbalanced data and can provide a distorted view of the effectiveness of the algorithm.

B.5. Types of models

In the current field of prognostic research a large number of models are available. In this chapter this will be categorized and explained in a critical way. The first division is in terms of the main ways researchers tend to perform research. Firstly, so called expert knowledge system are available. Secondly, statistical and stochastic systems. Thirdly, a very large amount of Artificial Intelligence methods. Finally, the physical models. Sikorska et al. [23] provides the basis of this division. However, the collection of artificial neural network will be expended to the complete field of AI, since recent research also provides great opportunities in terms of unsupervised learning and other AI techniques. Neural Networks is only a very small part of artificial intelligence [7].

KNOWLEDGE BASED MODELS

The first type of models are based on data that is given to the system by so called experts. A set of rules is made, where the status of the given system is predicted by these rules. This type of systems is popular in medicine. Given a set of symptoms, a patient has the following ailment. In general these system can be divided in expert and fuzzy systems [23].

An expert system creates rules in the form of IF-THEN. This system can only predict a discrete output and needs to be developed by many different experts. The system is however, simple to understand and can give valuable information.

However, this system has several weaknesses. First the system needs to be fully coherent. The rules defined do not contradict themselves, otherwise the system will give no useful answer. Secondly, the cost of updating, such is a system is very high, since rules need to be added and altered by an expert. An example of an expert system applied for fault diagnostics can be found in the research by Simeon et al. [22]. This study showed a reduction in maintenance cost and was a successful project. A fuzzy system is somewhat comparable to an expert system, The system the rules are based on a continuous scale. In terms of bearings maintenance this could be; If the vibrations are high and the temperature is very high then replace the bearing fast. A fuzzy system is in general more robust than an expert system. The fuzzy rules are mostly made by manual input, however a process called fuzzification is possible. The main strengths of fuzzy system is that a smaller set of rules is required than an expert system, however, it can work with noisy data and continuous sensors are available [23]. The weakness are however comparable to the expert system. An expert still needs to make the rules and the system needs to be updated.

An example of a fuzzy logic system is given by Majidian and Saidi [63]. In this research a Neural Network and a Fuzzy logic system are compared. The expected life of boiler tubes were predicted.

Expert system and fuzzy logic system are used in many different papers, however due to the complexity of the systems on board of aircraft and the lack of knowledge on the systems. This approach will not be effective for predicting RUL on the systems of a Boeing 787.

STATISTICAL AND STOCHASTIC MODELS

One of the most common models used in preventive maintenance in industry are the statistical and stochastic models. These models mostly apply observation from the past. This is also known as empirical knowledge. The data is used without understanding of the deeper physical behaviour of the component to develop a RUL prediction. The observation are fitted on a stochastic or probabilistic model [23].
A number of important statistical and stochastic models will be described in this section. Statistical models predict failure of a given component by analysing previous similar units and their failure rate and comparing it to current inspections. The simplest methods is trend evaluation. A single variable or sensor is linked to given value. If this critical value is reached. This critical value is set by analysis of statistical failure data. Another model, which is most commonly applied in industry is Proportional Hazard Modelling (PHM). This model uses no sensor data and only applies statistical failure data. PHM uses Time Between Failure, covariates and a baseline hazard function to predict failure based on failure observations. The advantage of this model is that no or low amounts of sensor data is required. The covariates of different parameters can tune the baseline hazard rate. This allows sensors or external data to impact the prediction. Older systems do not have these sensors or they do not recorded in a data base. Also PHM provides a statistical representation of failure. Other techniques such as most machine learning techniques provide only an accuracy with no confidence interval. However, this model requires the components to be IID (Independent and Identically Distributed). This means that every component has the same failure distribution and other component do not have influence on another. This also implies that every component has to be in the same condition when installed in the aircraft. This technique is for this research not optimal, since a high amount of sensor data is available.

Besides several stochastic models can be found for RUL modeling. Stochastic models can provide reliability features with respect to time. The most common one being the Mean Time Between Failure (MTBF) [23]. This also uses IID to create reliability estimates. However when low amounts of data are available these models can become pessimistic in terms of failure prediction. The simplest stochastic model, which is widely used in industry is the Aggregate Reliability Function (ARF). This model establishes a reliability function and analyses the failure rate at each point. The most common distribution is the Weibull function. It can fit normal distributed and exponential distributed data well. From this a decision can be made to replace the component at that time point. However, this technique results often in replacing components too early and result in increased cost. The largest group of stochastic models consist of Bayesian models. There is a very large group of models as described by Sikorksa [23]. One interesting group is the dynamic Bayesian networks, such as Markov and hidden-Markov models. Hidden-Markov models can use sensor data to change the model behaviour. However, for these type of models a large understanding is required of the failure modes and system dynamics. The data available does not give any insight to develop such a model.

MACHINE LEARNING

Artificial Intelligence is a very broad field of different techniques. It is in general a system that gathers inputs and uses these inputs to maximize its chance of reaching a certain goal. In current years types of machine learning and AI are very popular research topics. However, the used terminology is often difficult to find. As can be seen in Figure B.5, AI compromises the complete domain of techniques. Machine learning is one of the most common types used in recent years in AI. However, the earlier named expert systems, robotics and natural language processing, etc. are all part of AI. It is a vastly difficult and rapidly expanding field of research.

This chapter is going to expand mostly on the fields of machine learning, since the other field of AI are not as applicable for RUL prediction. Machine learning is divided into supervised, unsupervised and reinforcement learning. The types of models that are applicable to these techniques are explained, including their strengths and weaknesses. The main difference in these type of machine learning is the amount of feedback that is given [7].

Unsupervised learning Unsupervised learning is a technique to learn certain data patterns even though no required output is given. This is, predicting the failure of a component without a failure even happening or this data is never recorded. However, this component was closely monitored and a number of sensor parameters are available. Because no optimization can be made between the different features and parameters another approach is required [7, 32].

These approaches reduce the complexity of the data. In this way the data can be analysed in a more convenient approach. An overview of most unsupervised learning techniques is given by Dy et al. ad Hinton et al. [32, 64]. Unfortunately, no scientific article does give a complete overview of the types of unsupervised learning.

Clustering is a technique where the data is categorised a number of clusters. These

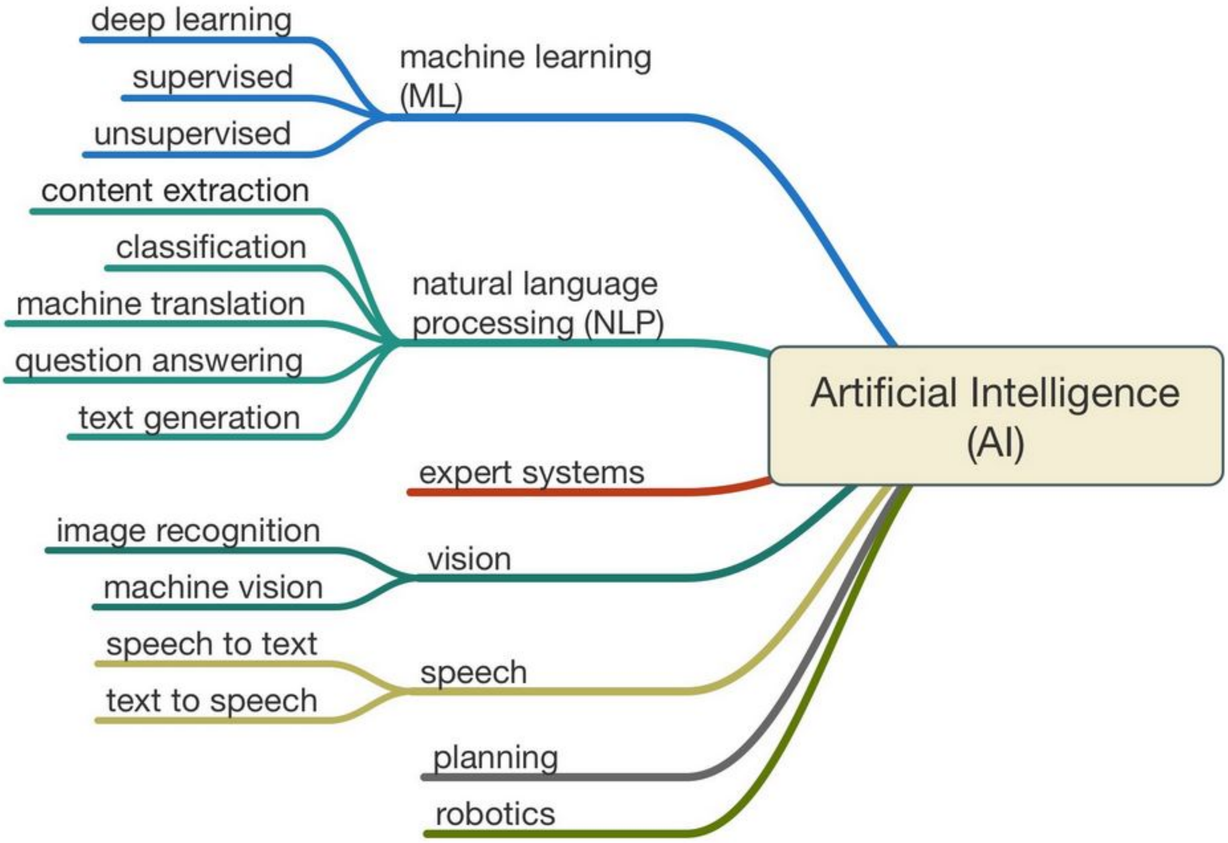


Figure B.5: Overview of different AI techniques and its global structure [4]

clusters will follow naturally from the data if they exist. The clusters can contain patterns or a structure type and can be used for example, in prognostics to already find anomalies in the data. These anomalies could lead to failure of the component and indicate that the life time of the component is decreasing.
Common techniques for clustering are:

- **Hierarchical clustering:** A clustering technique where the data is first separated in different clusters. The data will slowly get closer to one single cluster.
- **K nearest neighbours:** This clustering technique simply combines close values based on similarity. However, this model is slow on large data set and the distance, which can be set in the model is required to be set optimally.

Other algorithms could also be applied such as anomaly detection or latent variable models. Latent variables are a combination of different features. This is also known in terms of prognostics as a Virtual Health Indicator. For example, a widely used latent variable model is Principle Component Analysis. PCA reduces the highly dimensional data by selecting the features with the highest variance. This results in a simpler data set, which still shows the most important behaviour.
Finally a set of Neural Network applications are available such as auto-encoders and deep belief networks. These networks also aim to reduce the data or to cluster the given data.

Unsupervised learning can be used to explore the data in an efficient way and see if there are features, which could lead to a given failure mode in RUL. Simple forms of unsupervised learning are clustering and latent variable models. These models can be implemented easily and can show degradation behaviour in classes. For RUL prediction latent variable models are more appropriate, since a large number of features can be extracted from the data. Some data is not as important as other and these techniques work well with this information.

Unsupervised Neural Network techniques can also be applied for RUL prediction. For example, in [65–67] a LSTM network is used to detect anomalies or create a Health Index. These techniques can be of great interest, since they are already applied on different complex engineering data sets. This could give good results, especially for components where the failures have never occurred. However, this system is better for anomaly detection than for precise RUL prediction. If no failure data is present it is hard to predict a components life time by unsupervised learning.

Supervised learning For supervised learning, also the output of the model is given in combination with the input. The algorithm is creating a model, which links the given input to the output. The model is updated by means of optimization. For example, in prognostics this can be used if sensor data is available, which carefully monitors a set of parameters (input) and on the other hand, the moment of failure or component states are known (output) [7].
The general division of supervised learning techniques is regression, classification and techniques, which can do both. These models are further described together with the most common learning techniques used in theory.

Supervised Classification model These models are most common in predictive and preventive maintenance. These models typically indicate the life cycle of a part or component in a number of regressions or Health Stages as discussed in Section B.4. The model is predicting in which state the component is present [16]. This can be applied for anomaly detection: Is the component working as intended? Or this can be used for example, in different regions: One month to failure, one week to failure. The latter one can be of interest in RUL predictions, since necessity to perform maintenance increases.

There are several classification models available such as hard threshold, Support Vector Machine (SVM). K-nearest neighbours and logistic regressions [7, 31]. Hard threshold can be simply made defining a given features limit. If the given feature exceeds a given value a different category is reached. Most commonly this is done by making a decision boundary. This is a line or plane, which divides the different classes.
SVM is one of the easiest and most popular technique for classification models. This technique divides the data in two classes by a separating hyper plane or line. This can normally only be applied to linear data, since a linear division needs to be made between the two classes. To be able to work with two dimensional data a kernel function can be applied. This increases the dimension of data. In a higher dimension it might be possible to divide the data in different classifications. The strengths of this model is that it can model higher dimensional data. Also the model is quite robust against over-fitting. SVM's also have a weakness when it comes to larger data sets. The memory used in processing increases drastically in larger data sets. Therefore applying SVM on prognostics in newer aircraft would be not as favourable. Another widely used model is the nearest neighbour models. With this technique no boundary is used to classify data. This technique evaluates a new data set or point with already present "labelled" data points. These labeled data points

are already known to be of which class. The new point is classified by evaluating all the points closest around it up to k. For example, if k is six, four data points might be of class A and two of class B. This will result in the new data point being of class A. The strength of this model is mainly that it is easy to implement and is very flexible. However, useful features need to be engineered to work well. The weakness of this system is that the different stages or clusters need to be set appropriately and if the data does not show any information in terms of distance with one another, results in poor accuracy. Finally a logistic regression maps the function on a logistic function with a binary dependent variable. Only two different classes or HS can be defined with this method. The logistic function results in a probability function, which can be used for further analysis. For maintenance planning this is a useful tool, since most maintenance policies are based on a probabilistic model. this logistic function can only capture linear data, which is unfortunate for complex data.

Supervised regression models For RUL prediction regression models are not commonly used. Most often classification models are applied in literature. The models that can be used for regression can often also be applied in terms of classification. The simplest variation for regression is curve fitting. This can be done for a linear line, but also for higher order functions. Test data is used to fit a line through the test data points. New data points can now be evaluated with the line to predict a given value or RUL [7]. This method is one of the easiest to apply, since this requires only a fitting technique. However, the function to fit needs to be selected by hand. If a wrong function is used, a sub-optimal prediction is given. The largest disadvantage of this model is that this function can only capture a low number of variables. Multi-sensor input is hard to model and therefore not suited for complex aircraft data systems. Very useful and widely used regression models are random forest and neural networks. These models are discussed in detail in Figure B.5 and sec:machinelearing.

Random Forest Random Forest (RF) is an ensemble learning method. In ensemble learning a combination of hypotheses or decisions result in a given prediction [7, 68]. Instead of a single decision system as described earlier. The model is divided in a number of decision trees. Each tree divides the data based on a self selected set of features. Each step a new decision is made to continue on a certain branch. At the end of the tree a prediction is given. A RF is a combination of these decision trees. All the answers of the different decision trees are combined to get a better final answer. For supervised learning a type of training is required to predict the correct value. For RF this training procedure is bootstrap aggregating or "bagging". This reduces the variance of the model, while still maintaining the same bias.

The strengths of RF is that it requires relatively small amount of data points to be able to predict good results. This can be useful when low amounts of data a available. Another advantage is that a RF is less complex and more understandable than ,for example, a Neural Network. The features and decisions made by the model are retractable and can be explained. RF has a strength in terms of computational time. To train this model a low amount of time is required to create a working model. Besides a RF is robust against over fitting. This is a process where too many training data can result in over fitting. Over fitting is a process, where the model is only able to distinguish the training data, whilst the validation data is predicted less accurate. Finally, RF is able to rank the "usefulness" of the used features by providing a score. The features that provide the strongest relation to predict the best answer can be provided. This technique and unsupervised learning models can give insights in the features that lead to certain predictions.

RF also have a number of disadvantages. A larger amounts of data will not result in a further optimization. Other models, such as NN, can make use of larger amount data points. The features required to build the decision trees are required to be engineered and selected manually. If the wrong features are selected the model will result in weaker predictions.

Research using RF for RUL prediction is applied using experimental or simulated data [68, 69]. Wu et al uses a RF machine learning technique to predict tool wear. A large number of different features are used for cutting force, vibrations and acoustic emissions. In this article the RF modeled outperformed a vanilla Neural Network and a SVM. However, in this paper a purely experimental data set is used, which is often the case for these types of research. Also the type of NN applied is vastly outdated for 2017. Complexer NN might outperform the RF in this study. Other studies which apply RF is E. IJzermans [27]. IJzermans uses a RF on predicting failure for condition-based maintenance on the bleed air system of the Boeing 747. In this study a CNN is performing identical results as a RF. This study yet uses real operational data, which is closer to reality.

In conclusion, RF is widely applied model which is robust, easy to apply and compute and is able to indicate the strongest features. However, the features are required to be self engineered and selected. Also the model is not benefiting from extreme amounts of data which are sometimes available with the current amount of sensor data.

Neural Network An algorithm which in the recent years has gained a large amount of influence in prognostics and already applied often in data science is the Neural Network. In this section the different aspect regarding NN will be explained. Also more advanced techniques like Convolutional Neural Network, Recurrent Neural Network and Long-Short Term Memory are further described.

Neural Network: Neural Networks (NN) are algorithms, which consists of a selection of nodes. These nodes, also called neurons, are connected in layers as shown in Figure B.6. All the nodes are connected to all the nodes of the previous layer for a normal NN. There are in general two different types of models. First is the feed-forward network. The input is flowing from one end of the model in one direction to the output layer. The model itself has no internal state, only the weights of the network. Secondly, a recurrent network is available. This model is able to loop its output to its input. This allows the model to include previous inputs. In this way the model can "learn" from previous behaviour and adapt to it. However, this model is more complex and since it is a looping model, can result in convergence or divergence of the output. First the feed-forward model will be further explained, since it is the backbone of most of the NN models [7]. A complete overview of the different network types can been seen in Figure B.7.

Feed-forward NN The mathematical representation of a node is visualised in Figure D.2. In this node several input values (a_x) are given. These values are modified by a set of weights (w_{ij}), which are different for each link. In the model the weighted sum of the inputs in first calculated according to Equation B.5. This value is used in an activation function, which provides the output of the node. These activation is further described in Equation B.5.
The network itself is firstly consisting of an input layer. This input layer can be raw data but can also be engineered features (Section B.4). The neural network input is required to be normalized between 0 and 1. This is due to the nature of the activation functions, which are engineered to handle this type of data. After the input a layer a number of hidden layers are present. A vanilla-NN (or shallow-NN) is a network where only a single hidden layer is present, whilst a deep-NN consists of multiple layers. Deep learning has the advantage of self analysing important features of the data. The amount of layers and amount of nodes per layer are highly variable and are part of the hyper-parameters. Finally, a single or multiple outputs are given.
The outputs of the deep or vanilla neural networks are at the start not tuned and

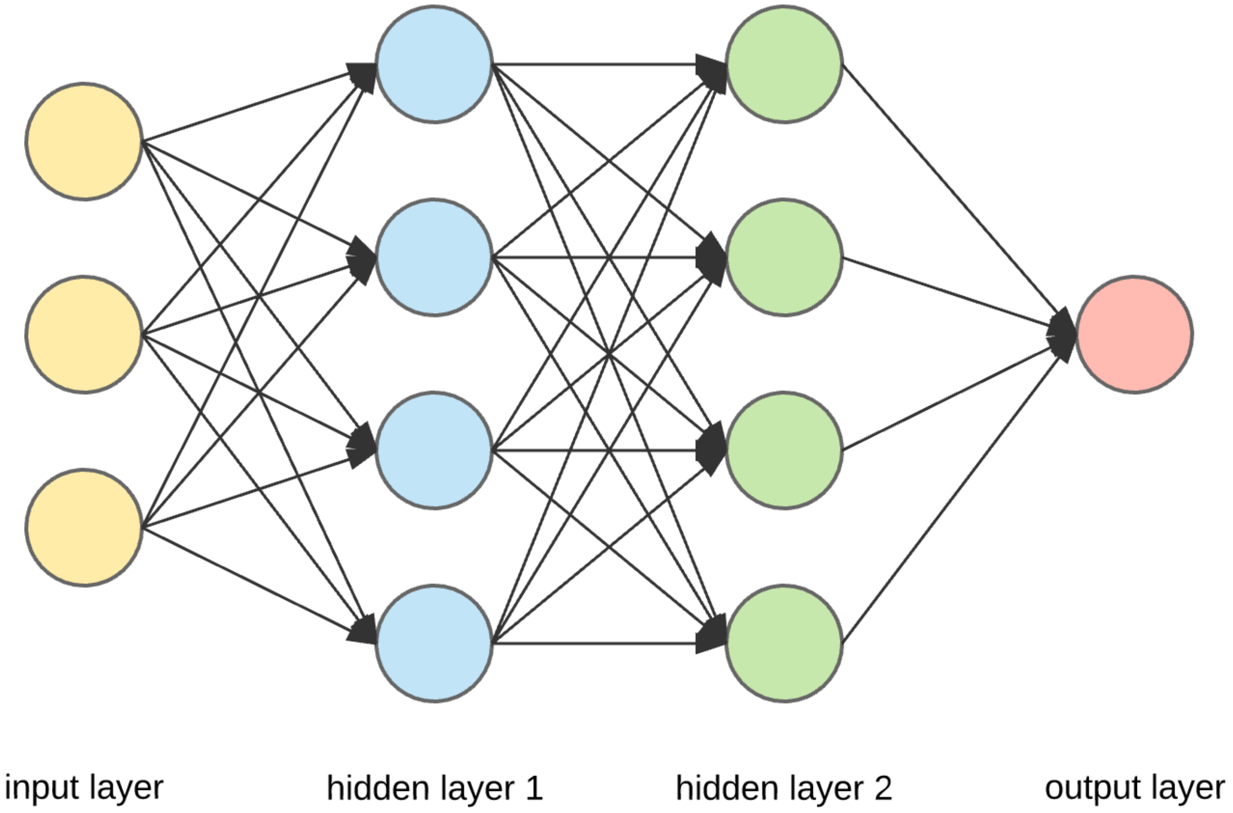


Figure B.6: A simple representation of a Neural Network. The layers are connected with each other from input to output [5].

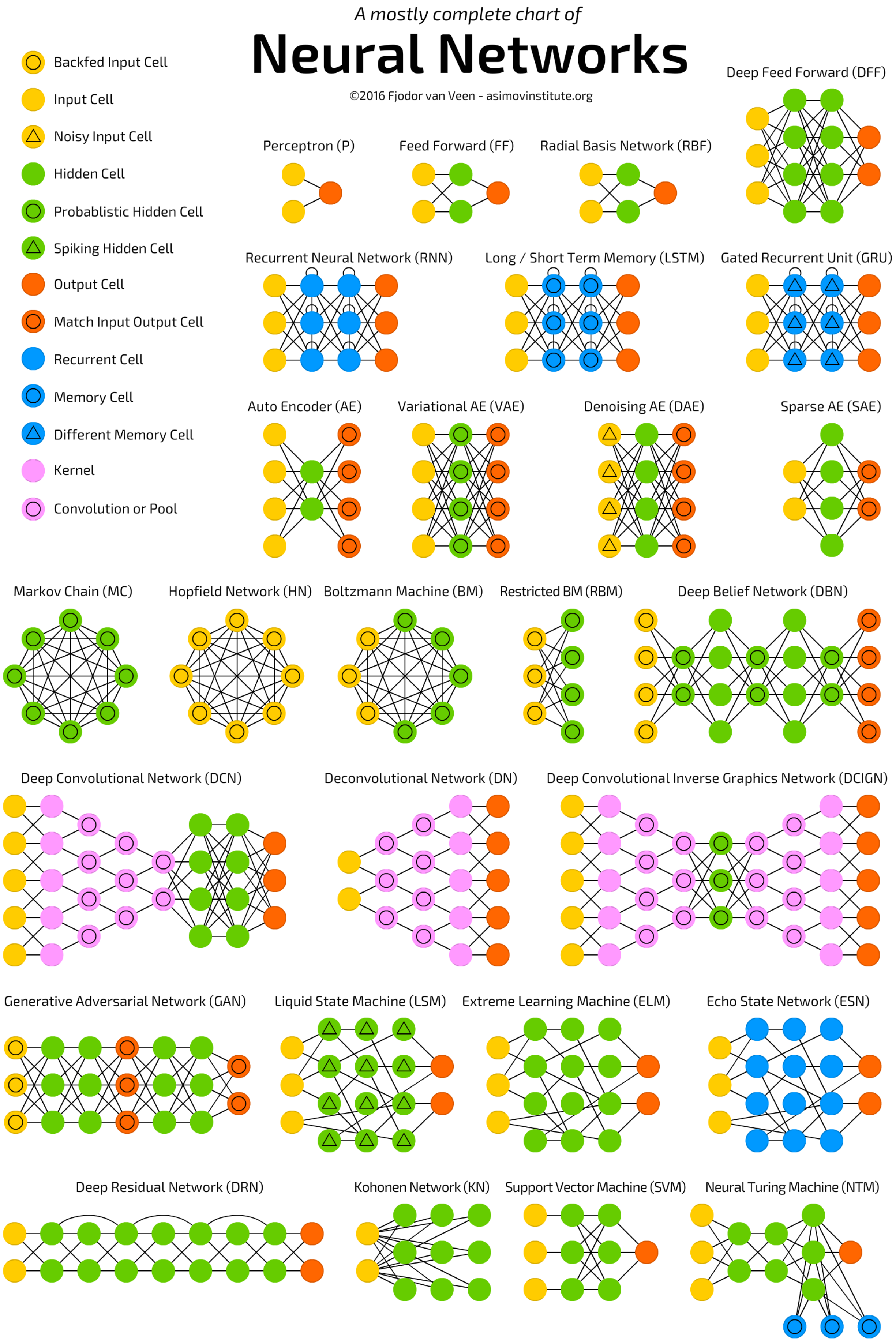

A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

Figure B.7: An overview of different NN and other techniques [6]

give random values. The weight of the model needs to be altered to provide the best output. This ’training’ is mostly applied using supervised learning, however unsupervised methods are available. In the supervised learning method, the output is verified with the reference data. The error between these two is used to alter the weight using a technique called back-propagation. This is in essential a more complicated gradient-descent optimization, where the weight change that results in the best output is selected. The weights are changed by a small amount, which is based on the learning rate. The learning procedure is repeated a number of epochs (cycles), until an optimum is reached.
The advantages of a vanilla feed-forward NN are that the system optimizes to provide the lowest error on the test data, nonetheless, still features need to be engineered to be truly effective. For deeper networks a new advantage can be used. Since in deep learning can identify critical features with its layer structure, feature engineering is not required. This also allows the advantage of large amount of data to be analysed and optimized. In recent years NN strategies provided good results in terms of predicting data.

However, Feed-forward NN and other NN models do have a number of disadvantages. Firstly the amount of data that is required to train the model appropriate is rather high. To less data leads to a problem called under-fitting. The model can not yet understand the data good enough to give valid predictions. From the requirements of large amount of data comes another problem. feed-forward NN are computational heavy and require sophisticated hardware and large amount of training time. The training time also increases, since an optimum of the solution is also hard to find, since there are many variables in the model selection (hyper-parameters). The hyper-parameters for NN models are also hard to interpret. The amount of nodes for example, do not directly propose a given parameter. For most other models the hyper-parameters represent real understandable features. Further explanation on hyper-parameters is in Equation B.5. Finally, NN techniques can be sensitive to over-fitting. The model can only predict the training data, but new data is predicted wrong. This problem can however be overcome by selecting a right amount of training epochs or setting a dedicated stopping mechanism.

$$in_j = \sum_{i=0}^{n} w_{i,j} a_i \tag{B.5}$$

Convolutional Neural Network A Convolutional Neural Network (CNN) uses a different architecture than only fully connected layers [34]. A fully connected layer is, where

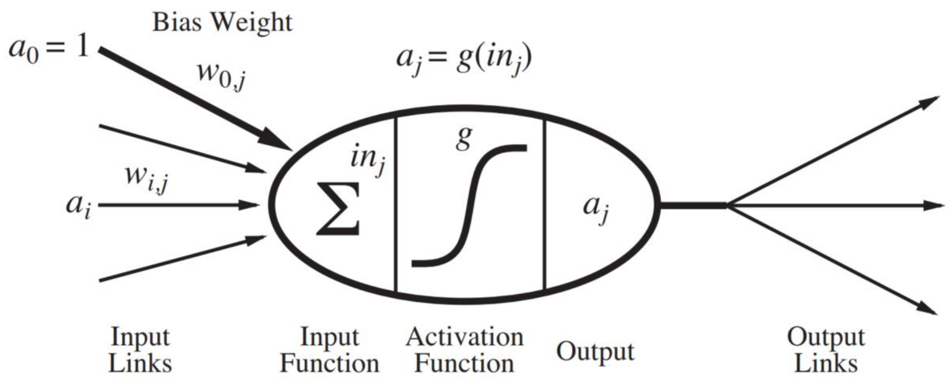


Figure B.8: A representation of the mathematical model of a neuron [7].

all the nodes are connected to the following layer. For a CNN a combination of convolutional layers and pooling layers are used. The convolutional layers divide the data in different features, which can be used for prediction. The pooling layers are used to reduce the amount of data and group close data points together. After each convolutional layer the data features are being more and more complex.

This technique is extremely useful for highly complex data recognition, where no simple features can be created by hand. Due to the convolutional layers, different features are extracted by the algorithm itself. In general CNN's are created for analysing highly correlated data. The location of the data is also not important, this is useful for picture recognition, since the same object is not always in the same spot. Finally, the pre-processing of data is very low, since raw data can be applied as the input. The weaknesses are however in terms of computational time. However, all NN systems does have this drawback and can be overcome using more sophisticated computers. Secondly, since the features are self selected by the model, results in extremely abstract features. These features can not be explained and result in a so called "black box" model.

In terms of applying CNN for RUL prediction different approach are available. Li et al uses a deep CNN to predict the RUL of the C-MAPSS data set [25]. As an input the different features are given in a matrix multiplied with the time sequence. So a two dimensional matrix is provided. Afterwards four convolution layers are placed. Afterwards two fully-connected layers are applied, instead of the earlier mentioned pooling layers. Finally, a drop-out technique is applied to prevent over fitting. The largest difficulty was the vast amount of hyper-parameters. The model was testen against different other NN techniques, such ass RNN and LSTM. It showed better

performance than the other networks. This paper shows a useful method for large, un-linear data sets. This model was applied to a simulated data-set so applying this to real data applications can be extremely useful. However, the comparison with other techniques is a bit doubtful, since completely optimising all these techniques with the right hyper-parameters is a highly time consuming work.
An earlier thesis of Erik IJzermans [27] uses a CNN to predict a Flight Deck Effect 10 days in advance using real life data. The input style is the same as Li, however, the optimization in terms of hyper-parameters is better described by means of a grid search. Also real data is used to predict the model and no experimental or simulated data sets are used.

Recurrent Neural Network In general RNN's are able to show dynamic behaviour due to the internal links between given nodes and are able to send information between time-steps [34]. This allows a RNN to keep states of earlier inputs to optimize its outcome. RNN's can be used to establish a prediction if a certain order in the inputs is useful. This is for example used in language and speech recognition, playing video games or music generation [70].

Recent research by Guo et al. [26] used a RNN to predict the RUL of experimentally tested bearings. A set of features where selected and applied in a RNN. With this a new RNN Health Inidicator is made, which increased the performance of the RUL prediction. This research shows the use of a RNN, however the research was only applied on an experimental set-up and the features are required to be still selected by hand. Also the primary goal of this paper is to create a more useful feature. Another article utilising a RNN to estimate RUL is by Felix Heimes on the IEEE challenge data set of 2008 and became 2nd [71]. In this paper several different methods are proposed. First a tapped delay line is introduced. Only with this input sequence it is hard to determine how much time data is needed and the amount of data input increases drastically. A RNN, however uses internal memory and can model more complex non-linear data. The downside of this is that the gradient of the weights of model. It now also has to take into account variations on earlier inputs, not only the current error. Heimes uses Back Propagation Through Time (BPTT) to compute the gradient. Finally, the weights are updated with an Extended Kalman Filter (EKF). With this procedure not all the data is required to update the weights and increases computational time. Finally an evolutionary algorithm is added to optimize the hyper-parameters of the model. This is an interesting approach to improve the accuracy of the data. This approach is however also applied on a nonreal life.

Long-Short Term Memory Neural Network LSTM is a different type of RNN. It uses memory cells with gates to save a given state over multiple input steps. When the gates are closed the given memory cell can not change it value. This results in the LSTM being able to retain information over more time-steps. In normal RNN's the information of earlier time-steps is normally lost [34]. LSTM networks are applied in many fields such as speech and handwriting recognition.

The different LSTM network nodes all have a connection with the previous layer and a connection with the previous time-step. The memory cell can let a cell keep its value.

Several research has been done in terms of LSTM application to RUL prediction. Zang et al. uses LSTM to predict the RUL of lithium-ion batteries [72]. A mean square back-propagation technique was used including dropout technique to prevent overfitting. The data-set was obtained by experimental data tests. The LSTM network was compared to a simple RNN and SVM. The LSTM performed better than both. Wu et al. applied a vanilla LSTM to predict the RUL of the NASA Ames turbofan engine data set [24]. The model showed promising results compared to other RNN models.

The two articles by Zang and Wu however both used experimental or simulated data as many researchers do. The downside of the use of RNN or LSTM is that in normal operation data gaps can occur. RNN and LSTM both require a complete string of data to obtain reliable data. If this is not the case a type of filtering is required or this method is not possible. In the data sets used by Zang and Wu completely valid data sets where used, without gaps. LSTM and RNN however pose a great feature in terms of RUL prediction. Failure behaviour in earlier samples can be used to finally predict a better failure prediction.

Finally a different type of approach in terms of using LSTM for unsupervised learning is being investigated by some researchers. Malhotra et al. uses a LSTM encoder-decoder algorithm for multi-sensor prognostics [66] based on earlier research [73]. Using this model normal behaviour can be filtered from close to failure behaviour. This type also requires a complete set of data-points across time. A downside of this technique with regards to RUL prediction is that this only can detect an anomaly.

Supervised NN models are able to give a continuous or more divided indication of RUL. A slightly later study of Malhotra provided a HI, created by the LSTM encoder-decoder algorithm that can provide a RUL estimation based on HI curve matching [67]. This technique can become useful in the future, since for some components a lot of sensor data is available, but no failure data is recorded. The LSTM can capture more difficult non-linear behaviour, than for example a SVM or PCA.

Hyper-parameters All the earlier named NN models all have a large number of different variables in terms of model size, type, learning method, activation function etc. These are called hyper-parameters. In this section the most important hyper-parameters are explained. Finally techniques, which help selecting these parameters are further discussed.

Type of hyper-parameters:

A large number of variables are available for working with NN's. First the amount of layers in the network are important. In general the more layers, the more complex models can be created. However, too many layers may lead to over-fitting and long computational times. The layer thickness is also variable. In most cases it is around the same size of the input layer.
For training purposes, the amount of training epochs and the error metric can be varied. Especially for unbalanced data the error metric can be essential, since optimizing for normal accuracy, might yield a sub-optimal result [27]. The amount of epochs need to be set in way that the important features in data can be learned enough, however prevent the algorithm from only being able to predict the test data. An optimum is required between over-fitting and under-fitting. This can be prevented by a basic number of epochs or a stopping algorithm. The steps that are used in the training are altered by the "learning rate". A high learning rate result in fast convergence when done appropriately. However, when it is too high the true optimum is not found and even divergence can occur. A small learning rate will converge slower to an optimum and can find a better optimum. The disadvantage of this, is that the rune-time is longer and the optimization might get stuck in a small local optimum.

Finally the type activation function can be changed. This function calculates the output of the different notes. The most common activation function are Sigmoid function and ReLu function.

Hyper-parameter selection: The hyper-parameters can be selected manually, but this is more than an art, than a science. Most of the variables are influencing each other and result in a sub-optimal model. There are however techniques to search for the right parameters [8].

First of all, a grid search can be done on several different features. The program can be run on every different combination and an optimal combination can be found. The weaknesses of this type of search is that this method requires an automated program, which self loops through all the combinations. Besides this program would take a long time to run. Another weakness is that NN's in general are converging to a certain optimum. However, each run might give a slight difference in end-result, due to a different starting point.

A more direct way of searching for the correct hyper-parameters is prescribed by Snoek et al. [35]. In this method Snoek uses a Gaussian process to find the optimal parameters. The technique provides an automated method, which surpasses human expert optimization and is applicable to many super vised learning techniques such as SVM and CNN.

Physical models

The last model type is a physical model of the component being researched. In this model a given system is modeled on macroscopic and microscopic scale to understand and predict the failure behaviour of the system [23]. This model consist of an equation or a number of equations from empirical data. This requires a lot of research and understanding of the system to be able to develop such a complex model. This results in this model being the most expensive and time consuming model to predict RUL.

There are many different examples of phyiscal models to predict failure behaviour of physical components. For example Ray and Sekhar used a stochastic model to develop a model, which prescribed the crack dynamics [74]. Since a stochastic model needs proper understanding of the behaviour of the system, results in many different test to understand the underlying physical behaviour. A more elaborated article by Dasgupta and Pecht describe more different mechanisms and models used to create physical models [75].

Most of the physical models research, is performed already 20 years ago. The techniques involving statistics and AI techniques are cheaper and easier to perform with the right data. Physical models are when applied correctly more accurate, but these techniques are only applicable to a single component. Other techniques are more flexible to apply to other systems.

Physical models for applying RUL prediction to complex aircraft components is extremely hard or even impossible. Also with the duration of a thesis of 6 months in mind would be to difficult and not giving the best solution. Therefore, physical models are not preferred for this thesis.

Sensitivity, processing and validation

Sensitivity analysis In the models used mostly three different aspects in terms of data quality and quantity were applicable. Firstly the amount of data. In most NN techniques a large amount of data is prescribed to obtain good results. It would be interesting to analyse how the amount of data influences the accuracy of the model. Secondly, is the continuous flow the data. In real world applications data samples are not always present for every time period [27]. This results for RNN and LSTM networks to be impractical. For sensitivity purposes data can be blanked out to see the result of more data gaps. Finally, is the data quality. The data can give several outliers or wrong sensor readings. This reduces the accuracy of the model. What can be researched is the effect of these error on the accuracy and its relation.

Data pre-processing The data to be used for RUL prediction is probably is someway affected by errors or data gaps. This can be be due to many different factors in sensoring or hardware. There are several data changes required before feeding it to the algorithm. These methods are mostly fairly small and very dependent on each data set, since most of the research is performed on carefully monitored data sets, no clear and concise article is written about all these error occurrences. For the supervised learning methods a critical step needs to be taken. This is linking the data to a reference label. For RUL predicting this means linking the data that resulted in failure to the actual failure. For this research it is quite important, since all the data can be grouped from placement till failure of a component or a multitude of classifications can be made already. This can be done by separating data from a week till failure, a month till failure etc. This allows for a non-binary classification model or even a regression model. IJzermans [27] uses a binary classification model where error prediction of 10 days is classified. By changing this margin or setting multiple margins could result in a better result.

Validation For validation an extreme large amount of metrics can be used to relate model performance. As earlier mentioned in Section B.4 normal metrics are sometimes not reliable when using imbalanced data. This has to be taken into account. An overview of more often used metrics for RUL prediction are described by Lei et al. [16]. Lei divides the type of RUL metrics in three categories. Depending on ground truth RUL, depending on run-to-failure data and finally, on live measurements. An interesting metric is the exponential transformed accuracy. This metric takes into account that overestimating the RUL is more costly than underestimating. This metric can also be used for regression modelling, since no confusion matrix is required. Finally, a number of online metrics are described by Hu et al. [76]. These metrics can be applied on continuous data problems for live RUL prediction.

Discussion on model types

In this chapter a large number of different models were explained. Knowledge based models are showing useful performance. However, due to the fact that no knowledge is failure knowledge of the system can be easily achieved, results in this method being unfit. Statistical and stochastic models are for maintenance predictions currently the leading standard. The models require simple input data and give probability output, which can be easily implemented. However, for complex symptoms, with many different failure modes. This becomes to difficult to model due to many assumptions. Therefore, this type of model is also not preferred in this research. Next is a large array of machine learning models. The primary division is made for supervised and unsupervised learning. For RUL prediction a supervised approach is preferred, since the failure data is available and unsupervised learning is able to easily provide a RUL. A RF is robust, requires less data and is showing less signs of overfitting. RF also is not a black box model and it can indicate which feature is the strongest. However, features are required to be build and other deep learning techniques can provide better accuracy.

NN models on the other can optimize on large amount of data and for convolutional networks even possible to extract own features. However, the model comes with a couple of downsides. First of all, a large amount of data is preferred. Secondly, the model has many hyper-parameters which need to be tuned accordingly. Finally, the model does not provide a probabilistic distribution. Other metrics need to be used to see the accuracy of the model. However, many different researchers used NN to experimental data with success. Also this model is compatible with more complex systems. The most promising model is LSTM NN, when the data is available. This can use data over multiple flights to take into account the current flight and previous flight to improve the RUL prediction. It might be useful to combine a CNN with

LSTM to find features and be able to capture them overtime.

C

DATA PRE-PROCESSING

Not yet graded

C.1. Data Set

The data sets used during this research are publicly accessible ($http://ti.arc.nasa.gov/project/prognostic-data-repository$) [15]. This set is made publicly available to boost prognostic research, by providing a clear and representative data set. This is a simulated data set, which contains four different data sub-sets. An overview of the different data-sets can be seen in Table C.1 [77].

Each data set is different in terms of training units, testing units, number of operational conditions and fault modes. The training units represent the amount of different engine life times are represented, which are from a given initial condition till failure. The testing units are a different set of engines, which can be applied for evaluating the model accuracy. The two operational conditions are High Pressure compressor (HPC) and turbo fan degradation. For FD001 and FD002 only HPC degradation is present, while for FD003 and FD004, both failure modes are present. FD001 and FD003 only provide data with one operating condition, which is at sea-level. FD002 and FD004 are operating at six different conditions.
Each set contains a N-by-26 matrix. Where N is the number of data points available. The first column represents the given engine id and the second value represents the operational cycle. The next three columns are operational settings of the engine. The last 21 columns are different sensor measurements provided by the engine. The different sensor description can be found in Table C.2.
The cycles represent the life-cycle of the engine. Although each engine is simulated with different initial conditions. This results in a varying lifetime of each engine and is represented by the number of cycles for that engine id for the training set. The last cycle for a given engine also represents failure or unhealthy. However, this is not the case for the test set. The test set is cut off at some random point. A document is provided, which gives the actual RUL prediction when the testing data is cut off.

This data is applied in many prognostic research fields and even after more than 10 years of availability, still inspire new research. However there are advantages and disadvantages to this set. The most prominent advantage is that new results can be easily verified with earlier research and results can be benchmarked. Another advantage is that the dataset requires no data cleaning, hence less data pre-processing is required and all the data points are available. The latter is important for time-series dependent algorithms, such as Recurring Neural Network (RNN) and LSTM. On the other, the CMAPSS dataset used is a simulated dataset, which also influenced by some form of noise. The data in reality might be different,showing different failure behaviour or displaying data-gaps. Therefore, more research is required on real complex systems. This is however difficult, since most real-life datasets are owned by commercial companies and are used for in-house use only.

Table C.1: CMAPSS data set overview

Hyper-parameter	FD001	FD002	FD003	FD004
Training Units	100	260	100	249
Testing Units	100	259	100	248
Operation condition	1	6	1	6
Fault modes	1	1	2	2

Table C.2: Sensor description of the C-MAPSS data set [2]

	Symbol	Description	Units	Trend
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 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	PCNR d md	Demanded corrected fan speed	rpm	~
20	W31	HPT coolant bleed	lbm/s	↓
21	W32	LPT coolant bleed	lbm/s	↓

C.2. SENSOR SELECTION

The CMAPSS dataset contains a total of 21 different sensor values. However some of these sensors show a monotonic trend in their output. Therefore, these sensors reduce the effectiveness of the prediction algorithm and only the sensors, which show a relevant variation are chosen. The technique used is mentioned by Zheng et al. and Li et al.[2, 78]. Earlier research used less different type of sensors, since these techniques were less able to deal with large amounts of data . For example Wang et al. uses a similarity-based approach with 9 sensors [79]. On the other hand some early deep learning research used all sensors [80], though recent research tend to reduce the dimensionality of the data by selecting the most potent sensors.

C.3. Ordering and Time-Window

The data obtained from the CMAPSS dataset is one long list of values. This data sets is required to be split up to accommodate each engine separately. This is required for applying the Time Window (TW) modifications and later validation of the results. The next step is to apply the time-window. A TW is achieved by sliding a frame over the given data. This frame is equal to number of selected sensors (n) by the length of the time window (tl). This TW is then placed one time step further to obtain the next data frame. This results in a number of data points equal to the length of the life span of the engine (ls) minus the Time Window length (tl) of data samples per engine (i.e. ls - tl). The label (the actual RUL of the engine at that cycle) is also stated for each data sample. This is required for later training of the algorithm. The first number of cycles, which are lower than the TW length, cannot be used for prediction. This results in a prediction gap during the first cycles. A representation of TW gap can be seen in Figure E.6a. Therefore, the method is used as explained in Section E.7 to combine good predictions, while still able to predict RUL after initiation of the prediction model.

Next the samples are divided in to m piece and is required to apply the DAG network by Li et al. [2]. This allows LSTM network to function for each different sample. The data samples are also transposed to link them to the LSTM and CNN network paths.

The LSTM is only applicable to the m pieces of data. This is required to prevent overfitting and wrongly updating of the model in the training phase. If the training sequence is kept in its original form, similar samples are together. A Neural Network will use that advantage to predict a given RUL and update the weight accordingly. In reality, the model is required to predict accurately at any given point.

C.4. RUL target function

To better model the failure behaviour of the engine, a piece-wise linear target-function is applied. This technique modifies the data by Equation C.1. TR is the Target RUL and c is the current cycle. This results in a RUL label representation as seen in Figure C.1. This technique was first applied by Heimes et al. [71] and divides the in a healthy stage and a degradation stage. When the system is healthy no or less abnormal sensor behaviour is visible. This results in a higher accuracy of the algorithm and further used by different researchers [2, 81, 82].

$$\mathrm{TR} = \begin{cases} \mathrm{c}, & c < 125 \\ 125, & c \geq 125 \end{cases} \tag{C.1}$$

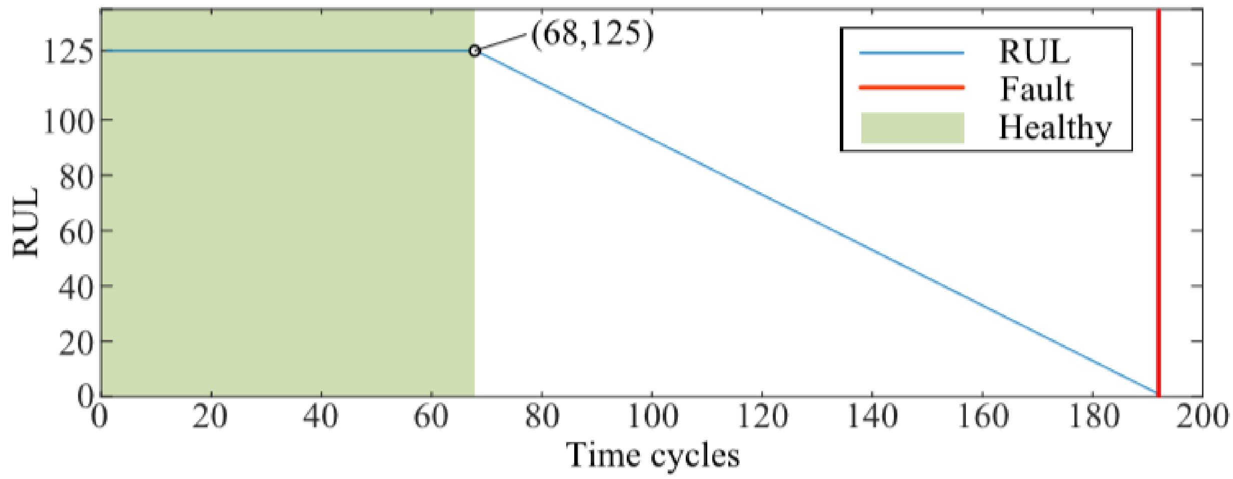


Figure C.1: Representation of the piece-wise linear RUL function [2].

C.5. NORMALIZATION

A Neural Network requires a normalized input to prevent exploding gradients [8]. The weight updates are too large and thus result in an inadequate solution. Therefore, a normalization of the data is applied before feeding it to the network. The two main versions of normalization are min-max normalization (C.2) and z-score normalization (C.3). Where min is the minimum value in the set for the given sensor, max the maximum value of the given feature, μ the mean value and σ the standard deviation [2].

Min-max normalization scales the data at the right scale, however outliers have a high impact. Z-score normalization does handles outliers well, however the scale of data can slightly change.

$$\frac{\text{value} - \text{min}}{\text{max} - \text{min}} \tag{C.2}$$

$$\frac{\text{value} \ - \mu}{\sigma} \tag{C.3}$$

C.6. DATA PRE-PROCESSING APPLICATION

- Select sensors required for prediction. The selection is based on the technique by Zheng et al. [78].
- Order the file per engine id. This is required for creating the Time Windows and for later validation.
- Create a new data frame, with the Time Windows applied. This results in a total of 17731 data samples for FD001 ($Ls - tl$). Resulting in $n \times tl \times \#samples$ data

points. Add to each samples the required RUL label.

- Divide each sample in m pieces and transpose the data. Resulting in $(n \times tl/m) \times m \times \#samples$ data points.
- Apply the RUL target function to the labels of the data.
- Normalize all sample data and labels with z-score normalization.
- Feed forward to the CNN and LSTM models.

D
Machine learning and network methodology

Not yet graded

D.1. Introduction

Machine learning is a complex, new and fast growing field of research. Many different concepts and theories are therefore required to apply these models. This appendix elaborates on supervised learning and the machine learning frame work in general. Afterwards the different network principles and structures are provided.

D.2. Supervised learning

Supervised learning is a method, where a given input is linked to a certain output, based on earlier shown examples of the correct input- output pairs [7, 69]. This can be achieved by applying several different algorithms such as linear regression, k-nearest neighbour, Random Forrest and Neural Networks. The general working principles of such algorithms is as follows. A model is chosen and adapted based on a set of training data. This is done in a way that by implementing the feature set (selection of training data) is representing the label (true value) as accurate as possible. This is often represented as the loss. When the loss is zero the output of the model is equal to the true value.

Supervised learning can be divided in two forms. A classification and regression model. The first mentioned is most used in literature [16, 27]. Classification is useful for CBM strategies, since impending failure can be classified. However, for predictive maintenance methods a regression model is required, since at any point a RUL prediction is required.

Overfitting is a key element when applying deep learning. Overfitting is the effect that occurs when too many training iterations are performed. This results in the model being only able to predict the training data. Other input data will result in inaccurate results. Therefore, the amount of training iterations (epochs) should be regulated and the final should be based on a separate training set.

D.3. Machine learning framework

Artificial Intelligence (AI) is a large field of research and inspires more research [83]. One of the most important branch of AI is machine learning. Machine learning consists of a large number of different techniques to predict or classify different data sources such as, data-sets, images, sound files and simulations. The two major groups of machine learning are supervised and un-supervised learning. In this research the focus is on the supervised learning.

Supervised learning is a method where a function is created, which maps its inputs to a given output. This is based on a set of example input-output pairs [83]. This

consists of many different algorithms such as Random Forrest (RF), linear/logistic regression, k-nearest neighbor, similarity training and Neural Networks [8, 83].
Another division can be made between regression and classification models. A classification algorithm couples the input to a given finite set of given "categories". The input is classified as one of the given "categories". This method is often used for image recognition, sound recognition or auto-correct functions. On the other hand, a regression algorithm couples the input to a given number (float). With this technique a final value is given, which best represents the example input-output pairs. [8, 16].
For the prediction of RUL of components, both techniques can be used. The health stage of the components can be divided in a set of different health qualities [16], which implies a classification model. Another method is to predict the amount of time/cycles till failure will occur and this results in a regression model. For this research the latter is chosen, since this allows for a better and more traceable prediction. Maintenance can be planned in advance and the status of the component can be checked on a regular basis. A classification model can suddenly switch from healthy to an unhealthy stage.

D.4. Neural Network

The applied CNN-LSTM network is a combination of both a Convolutional Neural Network (CNN) and a Long-Short Term Memory (LSTM) Network. Both of these networks are extension of the standard Neural Network (NN). First the NN is explained, afterwards CNN and LSTM are explained in Section D.5 and Section D.6.
A NN are originally inspired by the biological working of a brain [8]. It consists of a set of nodes (artificial neurons), which are connected in a set of layers, like synapses in a brain. A signal can be transferred between the different nodes and be used to create an input-output model. Each different connection has a certain weight. This weight influences the amount of importance a previous note has to the next note. These weights are changed during the training of the network, as explained in Appendix E.

A representation of a Neural Network can be seen in Figure D.1. A given input is given as the input layer. This input is then further propagated through to one or more hidden layers. Finally, the output of the last hidden layer is returned to a single or multiple output values. The mathematical notation is given by Equation D.1. Where a is the given activation at layer l and the j'th neuron. The activation function is represented as σ and defines how the input is transferred to a given output. There

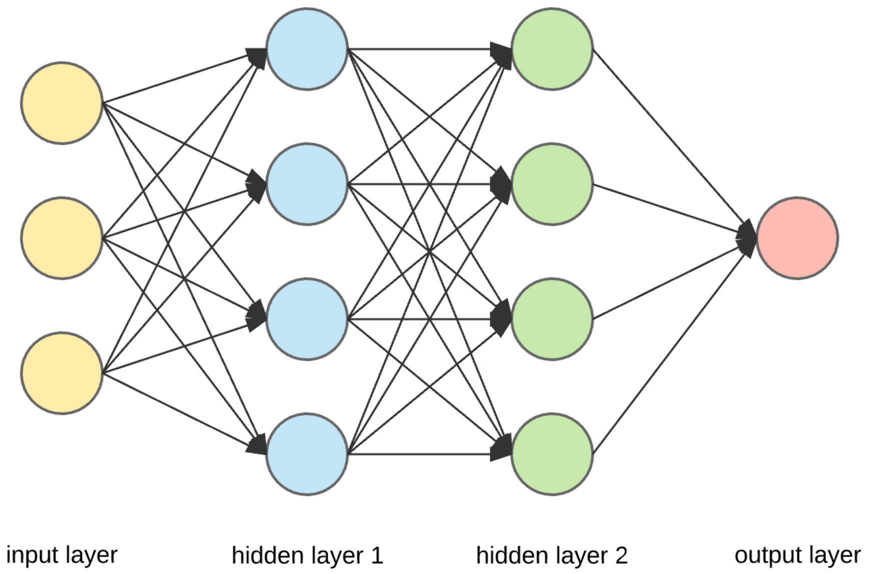


Figure D.1: A simple representation of a Neural Network. The layers are connected with each other from input to output [5].

are multiple activation function types and these are further explained and tested by Ramachandran et al. [84]. The weights are given by w and b the bias. A bias is used to shift the activation and better fit the actual data. Furthermore, this calculation need to be performed on all previous layers to obtain the current activation value. So the sum over all previous layers is required, k. This standard formula is applicable to most versions of neural networks.

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right) \tag{D.1}$$

D.5. CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (CNN) is mainly applied for image and sound recognition and creation. In most cases a higher dimensional data framework is processed by placing a filter over the data. This filter is known as a kernel, which can 'highlight' and 'specify' certain features in the data.

This allows for different features selection, than used in most statistical research. For example, in Figure D.3 an edge detection kernel is applied, which exaggerates the edges in the image. This is achieved by first transferring the left image in a 2D-matrix, where all the pixels are represented as a number between 1 and 256 (This number is dependent on the contract of the pixel). Afterwards a 3x3 kernel is placed over the data, with a specific value (e.g. Sobel Operator) [85]. Relevant information is highlighted to be able to create a higher accuracy for prediction.

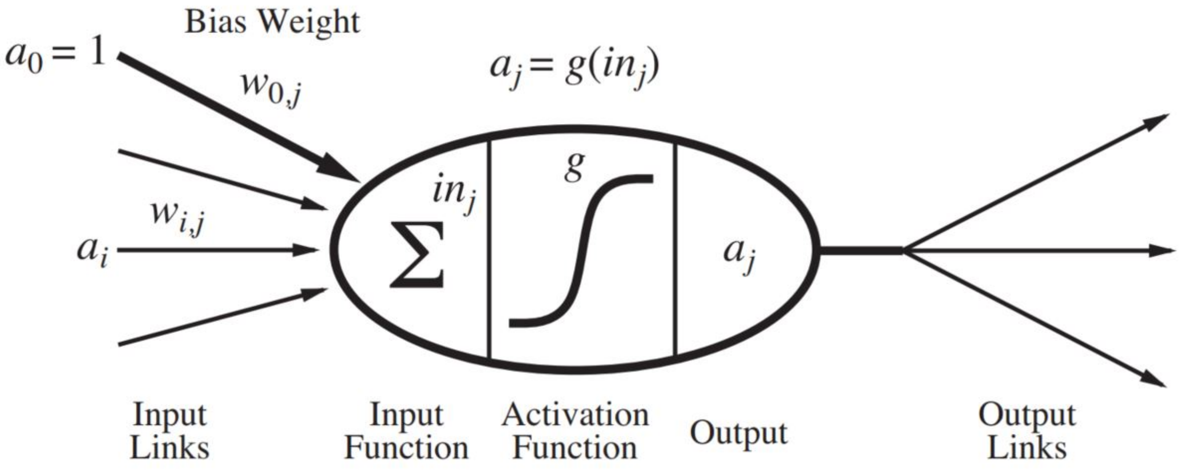


Figure D.2: A representation of the mathematical model of a neuron [7].

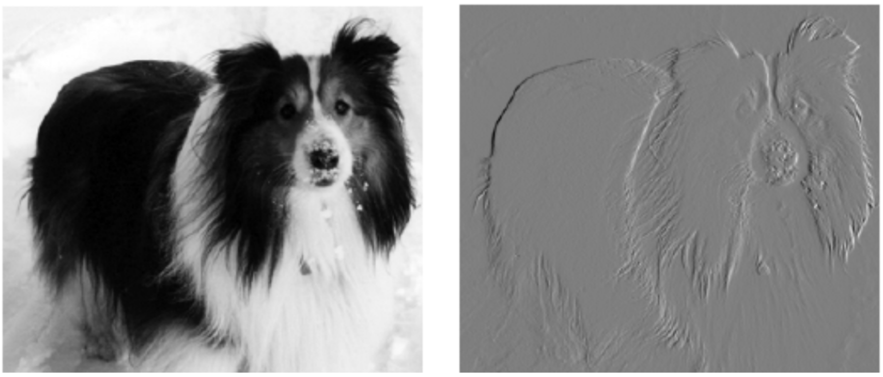


Figure D.3: Example of an edge detection algorithm [8].

In the CNN operation used in this paper is as follows. The data obtain from preprocessing (Over each piece of the data nk kernels are applied of size [k1,k2] with a stride of [s1,s2]. This operation reduces the data size to $m \times 6 \times 2$. A max-pooling operation is applied over the output of the first convolution. This max-pooling also applies a kernel like operation, except the largest value in this filter is the new value. This can be applied to reduce the size of the data. This pooling operation has a size of [p1,p2] and a stride of [ps1,ps2]. Finally, each kernel operation resulted in their own outcome (nk kernels). They are connected to each other, resulting in an output of 18. $(nk \times 6)$
The data output of the CNN is later combined with the data from the LSTM network, explained in Section D.7

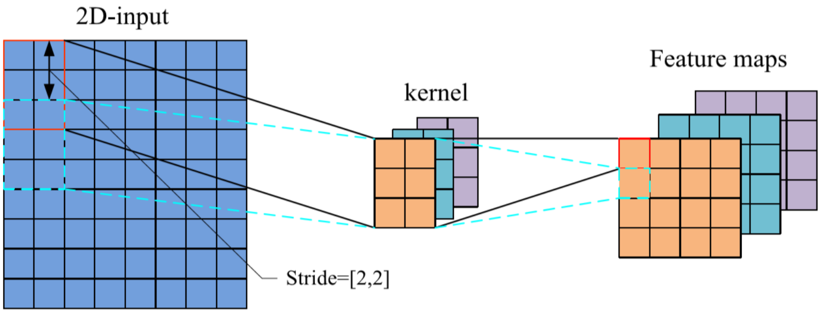


Figure D.4: Kernel operation on a two-dimensional input [2].

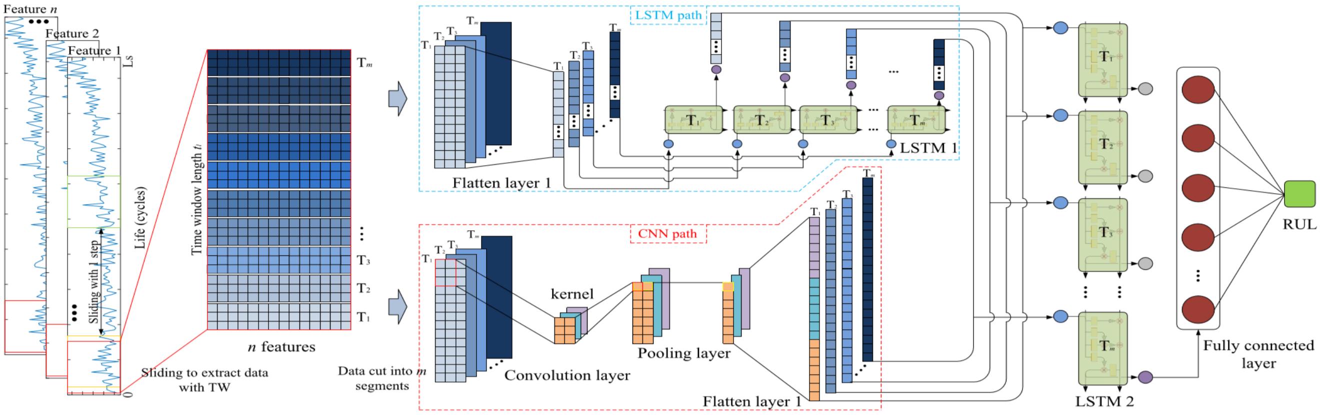


Figure D.5: A representation of the network architecture [2]

D.6. LSTM

LSTM is a type of neural network known as a Recurrent Neural Network (RNN). However, a LSTM is an extended version of a normal RNN, which was designed to overcome the vanishing gradient problem, encountered in a normal RNN. These networks are normally applied in situations where a sequence of data is available [86].

A vanilla LSTM unit contains a cell with an input gate, output gate and forget gate. The input gate regulates the amount of data that is introduced to the cell. The forget get gate regulates the amount of data that stays in the cell and the output gate is used to compute the output of the activation. The forget gate allows to store data over a larger amount of epoch/updates and thus allows for predictions based on a sequence of input data.

A limited selection of activation functions are available for a LSTM network. This network applies a Sigmoid activation function for its LSTM operations.

The applied LSTM path uses exactly the same input data and outputs the same size of output data as the CNN path. This is acquired by using a single layer with 18 nodes (layer size). The amount of layers in the LSTM dictates the outcome of the network. The number 18 is chosen to be able to directly combine the outcomes of both the CNN and LSTM path. The batch size is equal to three, since the total TW frame is cut into pieces with a length of 3.

A LSTM has fewer selectable options then a CNN and is only dictated by the input size, amount of layers, layer size and batch size. The embedding dimension is the batch size multiplied by the layer size in this example. The embedding dimension naturally flows from the layer size and batch size.

D.7. Combining

The CNN and LSTM networks discussed in Section D.5 and Section D.6 can be combined in a number of configurations. The serial combination is applied by Jayasinghe et al. [82]. The CNN operation is followed by the LSTM configuration and the outcome of the LSTM network is gained by a fully-connected network to obtain a single outcome (The RUL prediction).

The model applied in this research is a parallel combination and was explored by Dulaimi et al. and Li et al. [2, 81]. Both the CNN and LSTM path are executed simultaneous. The outcomes are afterwards combined to achieve a single RUL prediction. Li et al. sums the outcome of both networks and applies this in a second LSTM network. Where only the outcome of the last slice/piece is used to predict the final RUL prediction. However, this last slice is influenced by the earlier slices due to the LSTMs nature. This results in a more accurate prediction method than the serial network combination. The strengths of both networks are used in this way by the recognition power of CNN and the time series prediction of the LSTM. Both inserted with the same reference data.

E
Training and Results

Not yet graded

E.1. Introduction

This appendix elaborates on the training procedure and provides additional figures. Different figures are shown for normalization types, optimizers, learning rates and maximum RUL. The predictions behaviour is explained for different engine units and finally the different adaptations.

E.2. Normalization and Optimizers

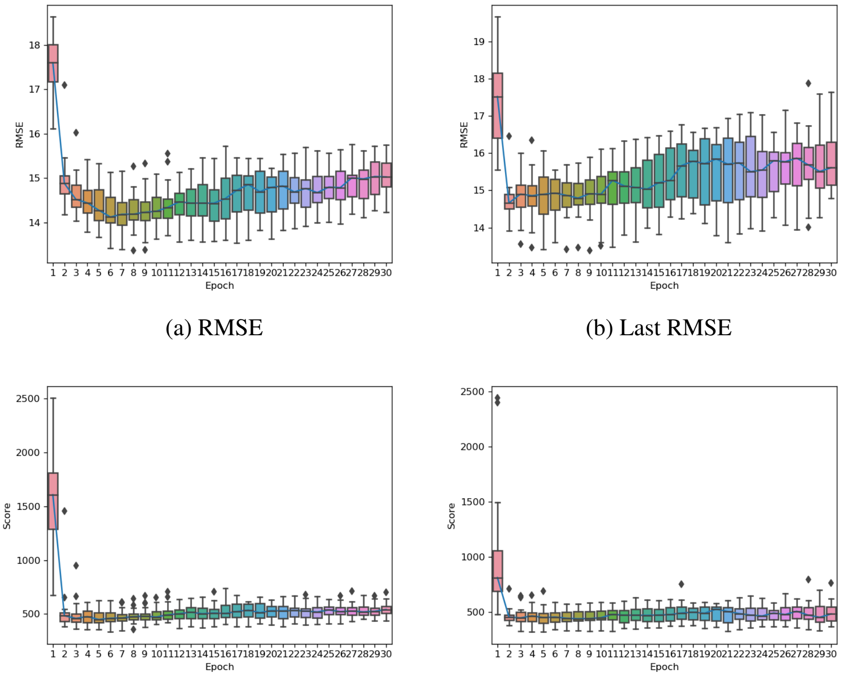
An overview of the different training progressions are given in Figure E.1, Figure E.2, Figure E.3 and Figure E.4. The figures are based on a combination of normalization type and optimizer. The other hyper-parameters are kept constant. For each combination a total of 4 sub-figures are represented. "a" shows the RMSE, "b" the RMSE based on the final prediction of each engine unit (Last RMSE). "c" depicts the score and "d" the last score.

The figures are based on training over the complete training set and after each epoch the accuracy metrics are shown based on the testing set. This is repeated for a total of 20 iterations and the results are plotted in a box plot representation with a line connecting the median values. The y-axis representing the given accuracy matrix and the x-axis the given epoch. The differences for accuracy metric, normal vs last and different combination of normalization and optimizer are further explained.

The difference between RMSE and score is most prominent in the difference in relative values between the first epoch and the later ones. While the RMSE is decreasing roughly from 20 to 14, is the score decreasing from 2000 to 400. Score decreases faster as the model becomes better in RUL prediction than RMSE. This is due to the way the score function is defined. The overall behaviour however, is reasonably similar and the figures are showing the same trend.

The differences between the ’normal’ accuracy and the ’last’ accuracy is also shows similar behaviour. However, the last prediction is diverging more in accuracy as the amount of epochs increase. This means that for prediction of the last cycle of the engine, training can result in a larger range of accuracy’s. This also leads to a larger amount of outliers.

The differences between the different normalization types can be seen when looking at Figure E.1 and Figure E.3. For Z-score normalization the accuracy decreases quickly and increases again after 8 epochs. The accuracy reduces slower for the min-max normalization. This probably due to the smaller variance in min-max normalization and the inability to prevent outliers from the data. RMS-prop has a better accuracy overall accuracy and a better optimum than the Adam optimizer. This data-set is most likely more suited for this data-set and for a different data-set this might not be the case.



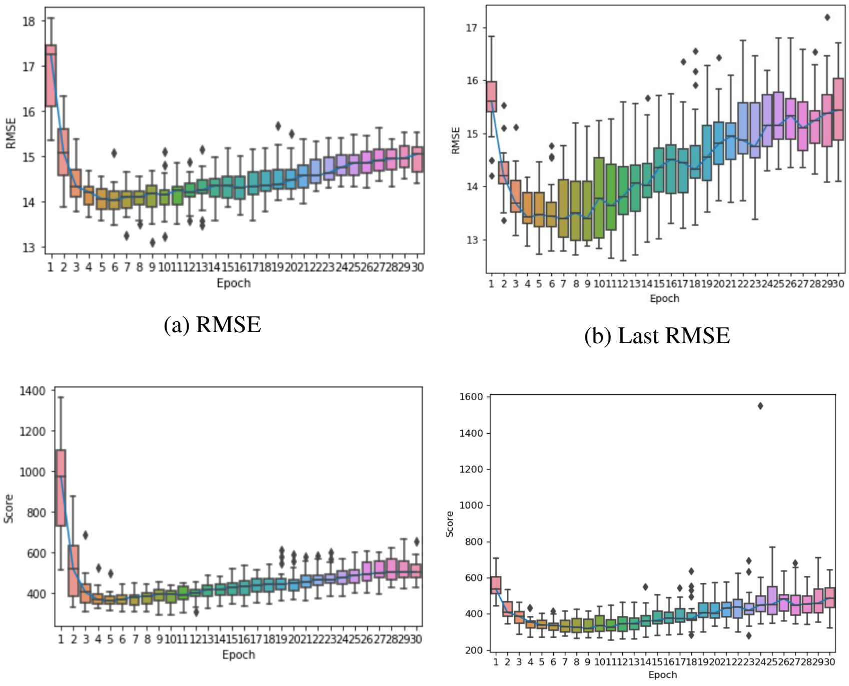
(a) RMSE

(b) Last RMSE

(c) Score

(d) Last Score

Figure E.1: Training accuracy with z-score normalization and Adam optimizer based on different training epochs



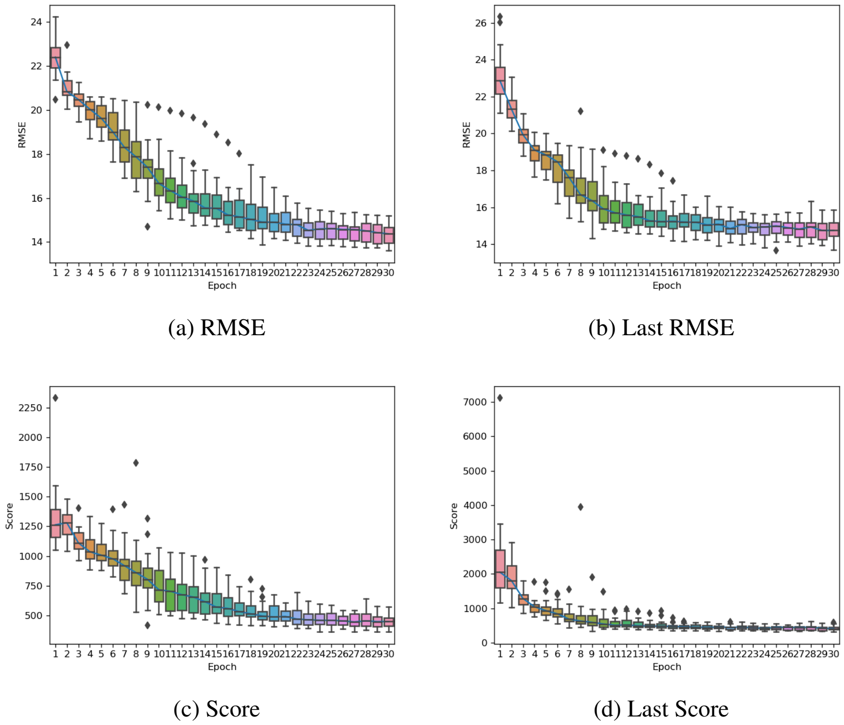
(a) RMSE

(b) Last RMSE

(c) Score

(d) Last Score

Figure E.2: Training accuracy with z-score normalization and RMS-prop optimizer based on different training epochs



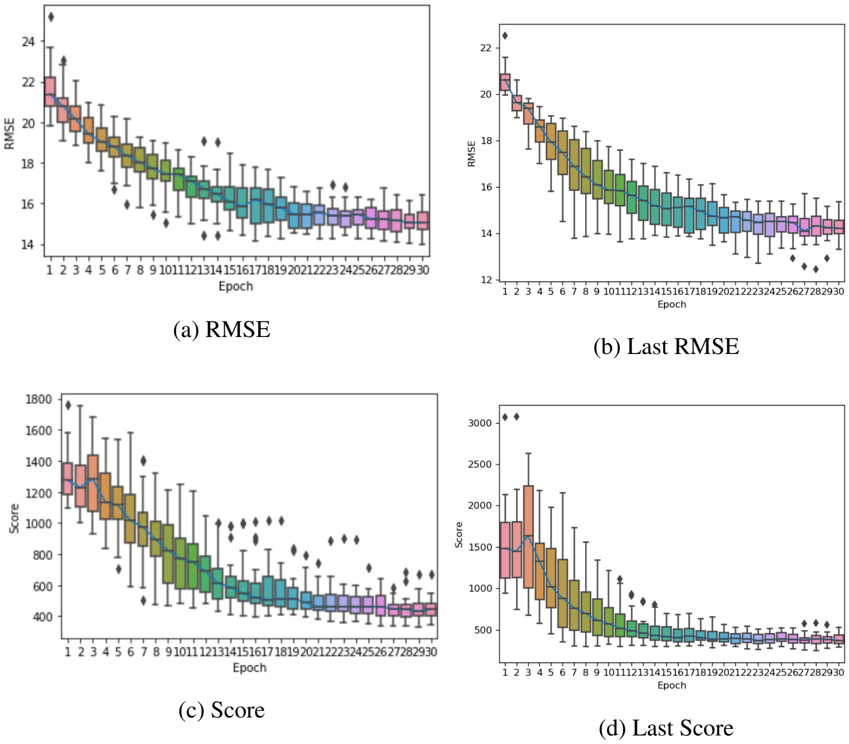
(a) RMSE

(b) Last RMSE

(c) Score

(d) Last Score

Figure E.3: Training accuracy with min-max normalization and Adam optimizer based on different training epochs



(a) RMSE

(b) Last RMSE

(c) Score

(d) Last Score

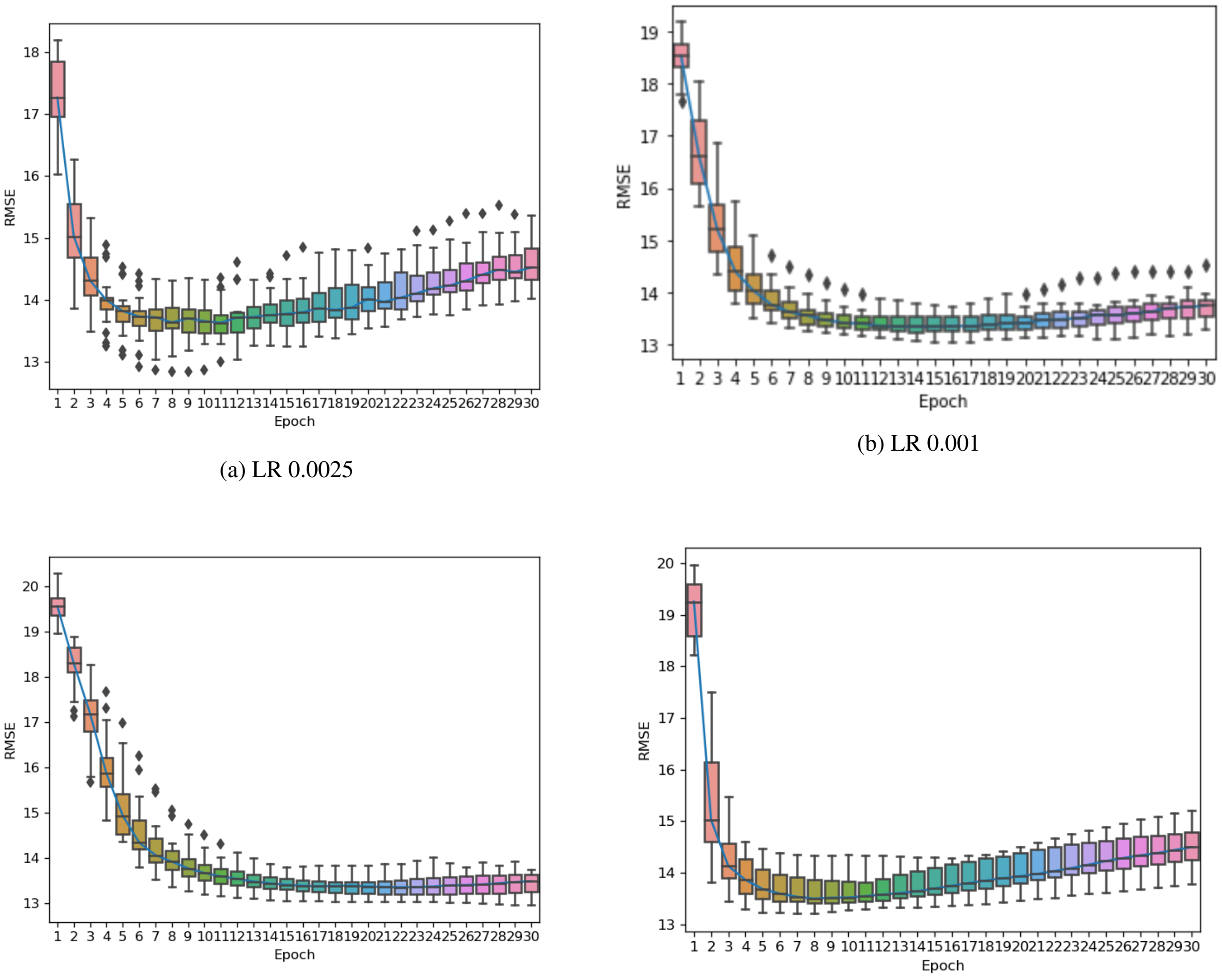
Figure E.4: Network accuracy with min-max normalization and RMS-prop optimizer based on different training epochs

E.3. Learning rate

The learning rate can be altered to change the update size after each mini-batch. For the RMS-prop and Adam optimizer, a dynamic updating approach for learning rates is applicable [87]. The learning rate is updated by optimizer according to the rate of loss measured of each update. In essence, the learning rate is actually an initial learning rate.

Different learning rates are shown in Figure E.5 with the RMS-prop optimizer and Z-score normalization. The optimal epoch is earlier achieved, when a high initial learning rate is used. The network training converges faster to an optimal point. However, the optimal point is moving forwards again when a very small learning rate is applied, due to the network being unable to find the true optimal point. To obtain the best optimum, a correct learning rate is required to be selected, which is not too high (optimal point is not found due to high step size) nor too low (optimization is stuck in a different local minimum).

As a side note, this technique is applicable for RMS-prop and Adam optimizers. For non-adaptive optimizers it is recommended to apply a different decrease in learning rate when training proceeds (e.g. decrease the learning rate every 1 epoch).



(a) LR 0.0025

(b) LR 0.001

(c) LR 0.0005

(d) LR 0.00025

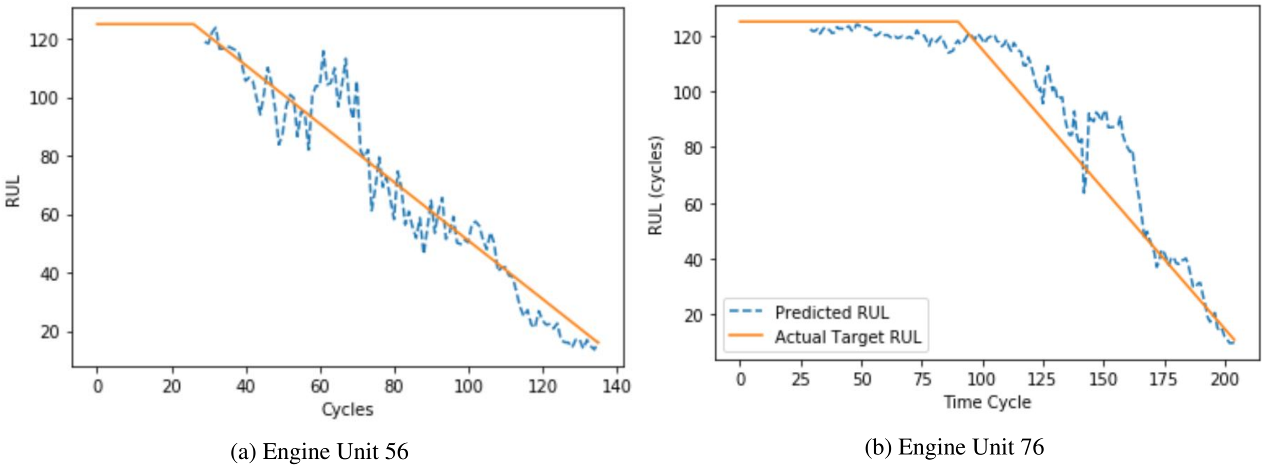
Figure E.5: Overview of the network accuracy based on different learning rates and training epochs

E.4. RUL Prediction

An overview of the prediction ability of the presented network with the given hyperparameters is shown in the following section. The results of six different testing engine units are presented in Figure E.6, E.7 and E.8. The first 29 time cycles do not have a prediction, since the time window length is 30. One line representing the Actual RUL and the other the predicted RUL by the algorithm. The predicted RUL should overlap with the actual RUL in the ideal situation.

Figure E.6 represents predictions, which have an accurate prediction over the complete data set. The prediction follows the general direction of the true RUL and thus results in a low error and high accuracy. The accuracy is best during healthy operation and when the degradation is severe. Around 50-100 cycles the prediction is less accurate and a larger deviation from the actual RUL can be observed. This might be caused by the effect that sensor data deviation are not as detectable in the

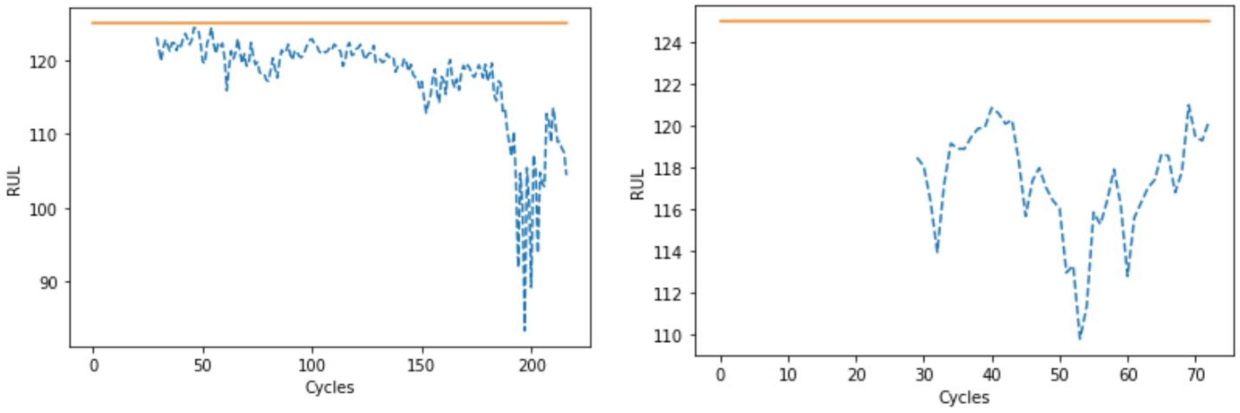
early failure stages as in the later stages.
Figure E.7 are engine units which only have testing samples above the maximum RUL. This should result in a prediction, which is always equal to the maximum allowable RUL (125). However, the prediction is mostly bellow the actual RUL. This is due to the model normalization, which divides the data between a certain range. For min-max normalization it is never possible to predict above the max RUL and for z-score normalization is rarely possible. Another aspect is that the model is predicting all possible data points. Resulting in under-predicting being mostly favourable.
Figure E.8 show engine units, which have a higher accuracy error. Sub-figure (a) indicates degradation patterns before actual degradation is present. This results in the degradation spikes in the early cycles. In sub-fire (b) the degradation is starting too early and is under-estimating the RUL for all data-points. These degradation patterns are not recognised correctly by the model and thus result in an inaccurate RUL prediction. These effects might disappear automatically as the accuracy of future models improves. Although, different effects might result in this behaviour, which is currently not captured by the model. Additional sensors, flight phase, type of flight, altitude and other operational/technical parameters might reduce the frequency of these errors.



(a) Engine Unit 56

(b) Engine Unit 76

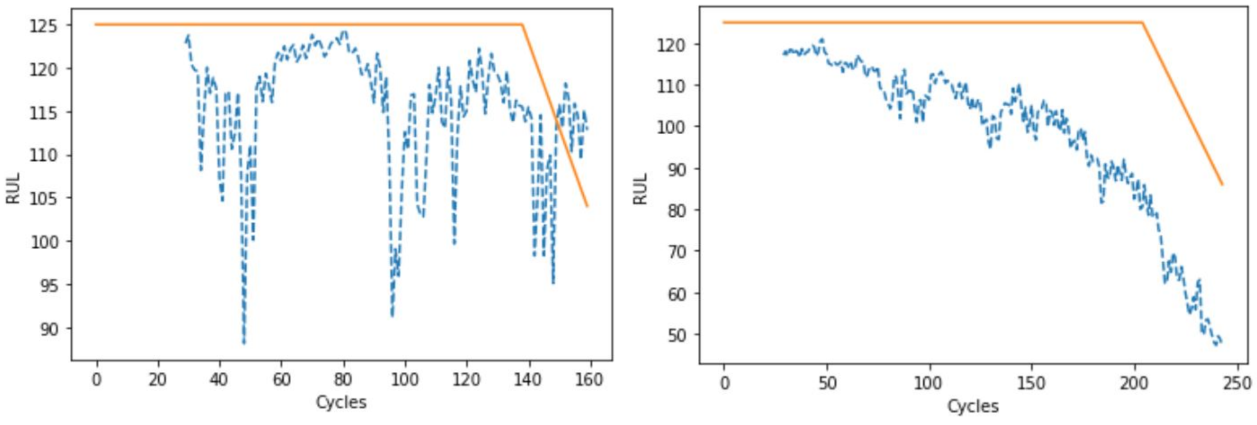
Figure E.6: Representation of two engine type predictions that show accurate predictions



(a) Engine 12

(b) Engine Unit 83

Figure E.7: Two RUL predictions for engine units that are in healthy state



(a) Engine Unit 57

(b) Engine Unit 93

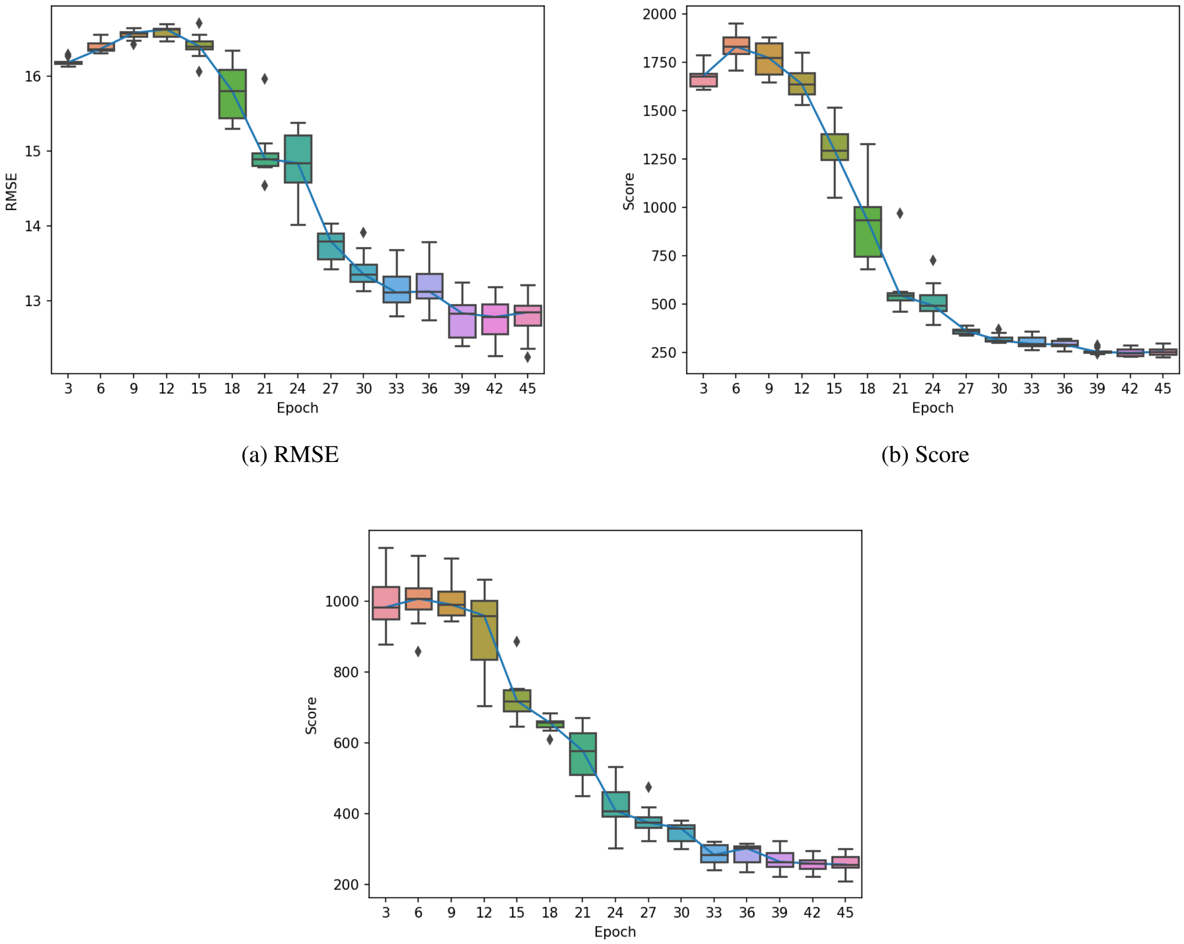
Figure E.8: Two predictions for engine units that are showing less accurate predictions

E.5. MAX RUL

Changing the maximum allowable RUL alters the accuracy of the network and the ability of predicting further in advance. The training and testing is re-scaled accordingly and used for training and validation respectively. The accuracy increases as the maximum allowable RUL reduces. For practical application it is advised to select the max RUL at a point where signalling early failure has an impact on operations and maintenance scheduling. If the maximum RUL is too low, it might not be useful anymore for planning maintenance operations and even dangerous in terms of safety. When a too high maximum RUL is adopted, results in too inaccurate predictions or it might not be necessary to know, since maintenance procedures are already at a higher interval.

E.6. ATW

An overview of the network accuracy of different time window lengths can be seen in Figure E.9. The overall shape of the figure is the same for all different accuracy metrics. The accuracy first slightly decreases and afterwards the accuracy increases until an optimum. What also can be seen is that the variation/range of the data for the middle sizes of TW (15-24) is larger than the other TW lengths. This is most likely due to the health stage most of these prediction are present and the respective accuracy and thus a larger variance. Shorter TW lengths and longer TW lengths are mostly present during healthy and degradation stage respectively. This allows for better predicting with a lower variance.



(a) RMSE

(b) Score

(c) Last Score

Figure E.9: Network accuracy for different time window lengths

E.7. SUB-NETWORKS

The sub-network technique is built on the idea that a single network is able to predict the complete range with an adequate accuracy. Nonetheless, a specifically trained network might improve the accuracy network when divided in dedicated sections/health stages.

The training data is divided in the dedicated health stages specified in Figure E.10. With a Upper Bound (UB) of 100 and a Lower Bound (LB) of 30. These three sets are then used to train the network a total of 10 iterations and 10 epochs. The network (setting) with the median accuracy for each different performance metric was selected for the final results.

The complete sub-network method relies on a primary prediction by the original network for the test data. This prediction is then used to label the given sample in one of the three health stage models. The last prediction is then made by the specific sub-network. This is performed for the complete testing data set.

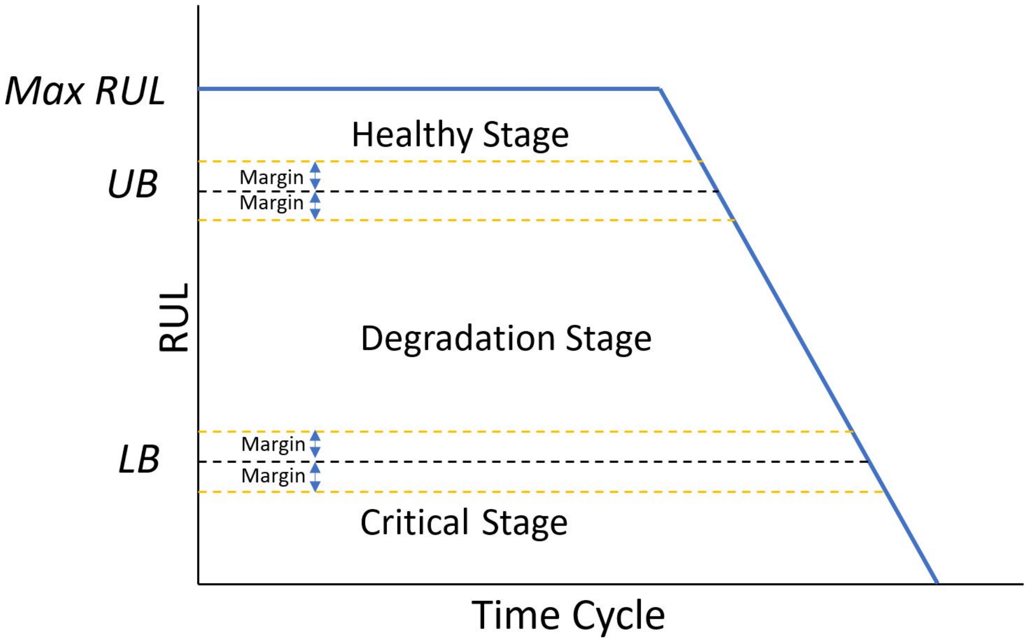


Figure E.10: Representation of the different health stages for sub-network training

The results of this network, however did not yet provided improvements in accuracy. The following improvements might be useful for further work and research.

- Altering the UB and LB to an optimum point. Also, dividing in two or more than 3 regions might be more optimal.
- Preventing false classification of the primary network by other strategies.

- First using a classification network as the primary network to classify the correct health stage. Afterwards the sub-network can be used.
- A different stopping condition (total number of epochs) for each type of sub-network. The different networks might have another point where the accuracy is optimal.
- Separate network configuration for each sub-network. However, this greatly increases the amount of variation in the network. Being computationally expensive.
- Using this technique with a different data-set, which contains more failure modes, such as FD003 and FD004 for the CMAPSS data set. This allows for a different type of sub-networking. The first primary network can classify the failure mode, while the sub-networks are trained on the specific failure modes. This can also be applied for different operating conditions. Although, the training set failure mode or operating condition should be available for each engine unit.

F

HYPER-PARAMETERS

Not yet graded

F.1. INTRODUCTION

The following sections gives an overview of all the hyper-parameters used and can be used as reference. The hyper-parameters are divided in three categories: pre-processing, network and training hyper-parameters. The actual values for each hyper-parameter can be found in table II and Table of III of the paper.

F.2. PRE-PROCESSING HYPER-PARAMETERS

- *Time Window:* The amount of time cycles used for each input to the model. The length can be changed to improve accuracy, however also influences the amount of available samples.
- *Selected Sensors:* A total of 14 sensors are selected from the 21 available. This is based on techniques applied by Zheng et al. and Li et al.[2, 78]. The TW lenght and amount of selected sensors dictate the size of the input.
- *Number of slices:* The data input ($TW \times sensors$) is divided in a number of slices. This is to implement the data in a LSTM, which requires a sequence of data.
- *Normalization:* The data is required to be scaled in a normalized version. This is to prevent the network from exploding its gradient and being unable to predict correct RULs.
- *RUL target function:* Introduced by Heimes et al. [71]. This technique uses a piece-wise linear function, appose to a linear RUL function. The maximum RUL in the label (target prediction) is restricted to a maximum value (Maximum RUL). This improves the accuracy of the network prediction by a better representation of the healthy stage and the degradation stage.
- *Maximum RUL:* The maximum RUL can be altered to change the RUL target function. A lower maximum RUL yields more accurate results, nonetheless early predictions are not able to predict RUL values higher than the maximum RUL.

F.3. NETWORK HYPER-PARAMETERS

- *Number of kernels:* One of the parameters for establishing a CNN network. It states how many different separate kernel operations are performed on one input. The output of each different kernel operation are combined and linearized to obtain one output.
- *Kernel size:* The kernel size consists of a width and a length and dictate how the convolutional network detects features. A larger kernel size combines data

from a larger range of data points. The size however is limited to initial size of the input data.

- *Stride:* It consist also of an x and y direction. The direction the kernel moves to calculate the output. A higher stride results in a smaller output size, since a larger portion of the data is skipped.
- *Padding:* A padding can be placed around the input data to prevent the data to reduce in size. Without padding the data always shrinks in size due to the required kernel size. There are a number of different padding operations, such as placing '0"s around the border of the input data (zero padding). However, in this research no padding is used.
- *Pooling size:* This consist of a width and length and is an operation similar to a kernel operation, however instead of calculating the output with weights, the highest, median, lowest, etc. is selected.
- *Pooling stride:* The direction in which the pooling operation is moving for each calculation.
- *LSTM nodes:* The LSTM nodes dictate the size of the LSTM network and directly influences the output of the network.

F.4. TRAINING HYPER-PARAMETERS

- *Training Epochs:* The total amount of training iterations over the complete training set. This is influenced by the learning rate, size of data, complexity and batch size. An optimal amount of training in epochs is required to prevent under- and over-fitting. The effect that the network is not yet able to recognize the data (under-fitting) or is only able to predict the training data and less accurate on the test set (over-fitting).
- *Batch size:* The batch size indicates the amount of data samples introduced in training process before each update, also referred as mini-batch. A larger batch size decreases computational time and results in less frequent updating. A smaller batch size takes more computational time, however prevents over-fitting, as the network is updated more often and no long term data recognition can be performed. For this model a mini-batch size of 100 is used.
- *Optimizer:* The optimizer influences the way the network is updated after each mini-batch.
- *Learning rate:* The learning rate influences the step size for the network update after each mini-batch.

- *Loss function:* The type of function, which calculates the loss before each update of the network. The loss dictates in which direction each weight is changed to improve the accuracy.
- *Activation function:* The activation function influences the output of each note by altering the output. This is to prevent the network from exploding its gradient and is key in deep learning.
- *Deep learning software:* Pytorch is the applied deep learning library for this research. It provides a toolbox for creating and running deep learning applications in Python. It is highly adaptable and especially applicable for research related work.

Bibliography

[1] M. A. Nielsen, *Neural Networks and Deep Learning*. 2015.

[2] J. Li, X. Li, and D. He, "A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction," *IEEE Access*, vol. 7, pp. 75464–75475, 2019.

[3] R. K. Mobley, *An introduction to predictive maintenance*. Butterworth-Heinemann, 2002.

[4] C. Kumar, "Artificial Intelligence: Definition, Types, Examples, Technologies," *Medium, https://medium.com/@chethankumargn/artificial-intelligence-definition-types-examples-technologies-962ea75c7b9b*, 2018.

[5] A. Dertat, "Applied Deep Learning - Part 1: Artificial Neural Networks," *Medium, accessed 05/11/2019*, vol. Towards Data Science, 2017.

[6] F. van Veen, "A mostly complete chart of Neural Networks," *asimovinstitute.org*, 2016.

[7] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[8] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," tech. rep.

[9] TU Delft, "TU Delft Information Literacy 3 course."

[10] NASA, "Goals and objectives for integated vehicle health management (IVHM)," *Report NASA-CR-192656*, 1992.

[11] R. Ahmad and S. Kamaruddin, "An overview of time-based and condition-based maintenance in industrial application," *Computers and Industrial Engineering*, vol. 63, no. 1, pp. 135–149, 2012.

[12] H. M. Elattar, H. K. Elminir, and A. M. Riad, "Prognostics: a literature review," *Complex & Intelligent Systems*, vol. 2, pp. 125–154, 6 2016.

[13] Upkeep, "Predictive vs Condition-Based Maintenance," *https://www.onupkeep.com/learning/maintenance-types/predictive-condition-based Accessed: 11/10/2019*.

[14] R. C. M. Yam, P. W. Tse, L. Li, and P. Tu, "Intelligent Predictive Decision Support System for Condition-Based Maintenance," tech. rep., 2001.

[15] Nasa, "Nasa Ames prognostic data reprository,"

[16] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 5 2018.

[17] O. F. Eker, F. Camci, and I. K. Jennions, "Major Challenges in Prognostics: Study on Benchmarking Prognostics Datasets," tech. rep.

[18] F. L. Greitzer and R. A. Pawlowski, "Embedded Health Monitoring Workshop," no. May, pp. 1–10, 2002.

[19] B. Bole, C. S. Kulkarni, and M. Daigle, "Adaptation of an Electrochemistry-based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use," tech. rep., 2014.

[20] B. Saha and K. Goebel, "Modeling Li-ion battery capacity depletion in a particle filtering framework," *Annual Conference of the Prognostics and Health Management Society, PHM 2009*, pp. 1–10, 2009.

[21] V. Giurgiutiu, "Tuned Lamb wave excitation and detection with piezoelectric wafer active sensors for structural health monitoring," 4 2005.

[22] E. A. Simeón, A. J. Álvares, and R. R. Gudwin, "AN EXPERT SYSTEM FOR FAULT DIAGNOSTICS IN CONDITION BASED MAINTENANCE," tech. rep., 2009.

[23] J. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing*, vol. 25, pp. 1803–1836, 7 2011.

[24] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, 1 2018.

[25] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering and System Safety*, vol. 172, pp. 1–11, 4 2018.

[26] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 5 2017.

[27] E.T. IJzermans, “Deep learning for Condition Based Maintenance,” *Master Thesis, Delft University of Technology*, 2018.

[28] E. Fumeo, L. Oneto, and D. Anguita, “Condition based maintenance in railway transportation systems based on big data streaming analysis,” in *Procedia Computer Science*, vol. 53, pp. 437–446, Elsevier B.V., 2015.

[29] J. P. C. Chiu and E. Nichols, “Named Entity Recognition with Bidirectional LSTM-CNNs,” tech. rep.

[30] H. Pan, X. He, S. Tang, and F. Meng, “An improved bearing fault diagnosis method using one-dimensional CNN and LSTM,” *Journal of Mechanical Engineering*, vol. 64, no. 7-8, pp. 443–452, 2018.

[31] Elite Data Science, “Modern Machine Learning Algorithms: Strenghts and Weaknesses,” *https://elitedatascience.com/machine-learning-algorithms, Acessed: 30/10/2019*.

[32] J. G. Dy and C. E. Brodley, “Feature selection for unsupervised learning,” *Journal of Machine Learning Research 5*, pp. 845–889, 2004.

[33] X.-Y. Liu, J. Wu, and Z.-H. Zhou, “Exploratory Under-Sampling for Class-Imbalance Learning,” tech. rep., 2006.

[34] A. Gibson and J. Patterson, *Deep Learning*. O’Reilly Media, Inc., 2017.

[35] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” tech. rep.

[36] Haibo He and E. Garcia, “Learning from Imbalanced Data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263–1284, 9 2009.

[37] TU Delft, “TU Delft libarary course information page.”

[38] J. S. Usher, A. H. Kamal, and W. H. Syed, “Cost optimal preventive maintenance and replacement scheduling,” tech. rep.

[39] M. Bebbington, C. D. Lai, and R. Zitikis, “A flexible Weibull extension,” *Reliability Engineering and System Safety*, vol. 92, pp. 719–726, 6 2007.

[40] T. Dohi, A. Ashioka, S. Osaki, and N. Kaio, “Repair-time limit replacement schedule Optimizing the repair-time limit replacement schedule with discounting and imperfect repair,” Tech. Rep. 1, 2001.

[41] T. D. Matteson, “Airline experience with reliability-centered maintenance,” *Nuclear Engineering and Design*, vol. 89, no. 2-3, pp. 385–390, 1985.

[42] ISO 13381-1, “ Condition Monitoring and Diagnostics of Machines – Prognostics – Part 1: General Guidelines: International Standards Organization,” 2014.

[43] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, “Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications,” *Mechanical Systems and Signal Processing*, vol. 42, pp. 314–334, 1 2014.

[44] A. Saxena, I. Roychoudhury, J. R. Celaya, S. Saha, B. Saha, and K. Goebel, “Requirements Specifications for Prognostics: An Overview,” tech. rep.

[45] M. Pecht and S. Kumar, “Data Analysis Approach for System Reliability, Diagnostics and Prognostics,” tech. rep.

[46] A. Abu-Hanna and P. J. Lucas, “Prognostic models in medicine. AI and statistical approaches.,” *Methods of information in medicine*, vol. 40, no. 1, pp. 1–5, 2001.

[47] W. C. Skamarock, J. B. Klemp, and J. Dudhia, “PROTOTYPES FOR THE WRF (WEATHER RESEARCH AND FORECASTING) MODEL,” tech. rep.

[48] A. Gertych, Z. Swiderska-Chadaj, Z. Ma, N. Ing, T. Markiewicz, S. Cierniak, H. Salemi, S. Guzman, A. E. Walts, and B. S. Knudsen, “Convolutional neural networks can accurately distinguish four histologic growth patterns of lung adenocarcinoma in digital slides,” *Scientific reports*, vol. 9, no. 1, p. 1483, 2019.

[49] C. Sbarufatti, M. Corbetta, A. Manes, and M. Giglio, “Sequential Monte-Carlo sampling based on a committee of artificial neural networks for posterior state estimation and residual lifetime prediction,” *International Journal of Fatigue*, vol. 83, pp. 10–23, 6 2015.

[50] A. Saxena and K. Goebel, “Phm08 challenge data set,” in *NASA Ames Prognostics Data Repository*, NASA Ames Research Center, 2008.

[51] T. Wang, J. Yu, D. Siegel, and J. Lee, “2008 INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems,” tech. rep., 2008.

[52] A. Mosallam, K. Medjaher, and N. Zerhouni, “Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction,” *Journal of Intelligent Manufacturing*, vol. 27, pp. 1037–1048, 10 2016.

[53] G. Prins, "Prognostics on aircraft components in the field of predictive maintenance," *Master Thesis, Delft University of Technology*, 2016.

[54] J. Lee, H. Qiu, G. Yu, and J. Lin, "Bearing Data Set, NASA Ames Prognostics Data Repository," 2007.

[55] L. Hu, N. Q. Hu, B. Fan, F. S. Gu, and X. Y. Zhang, "Modeling the relationship between vibration features and condition parameters using relevance vector machines for health monitoring of rolling element bearings under varying operation conditions," *Mathematical Problems in Engineering*, vol. 2015, 2 2015.

[56] M. Fordellone, A. Bellincontro, and F. Mencarelli, "Partial least squares discriminant analysis: A dimensionality reduction method to classify hyperspectral data," pp. 1–24, 2018.

[57] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," tech. rep., 2002.

[58] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 2838, pp. 107–119, 2003.

[59] M. Kukar and I. Kononenko, "Cost-Sensitive Learning with Neural Networks," tech. rep.

[60] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 6 2006.

[61] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," tech. rep.

[62] C. Drummond and R. C. Holte, "Cost curves: An improved method for visualizing classifier performance," *Machine Learning*, vol. 65, pp. 95–130, 10 2006.

[63] A. Majidian and M. H. Saidi, "Comparison of Fuzzy logic and Neural Network in life prediction of boiler tubes," *International Journal of Fatigue*, vol. 29, pp. 489–498, 3 2007.

[64] G. Hinton and T. Sejnowski, *Unsupervised Learning: Foundations of Neural Computation*. MIT Press, 1999.

[65] M. Verleysen, Universite Catholique de Louvain, Katholieke Universiteit Leuven, C. I. European Symposium on Artificial Neural Networks, M. L. . .-. Bruges, and ESANN 23 2015.04.23-25 Bruges, *Proceedings / 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2015, Bruges, Belgium, April 22-23-24, 2015*. Ciaco, 2015.

[66] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection," 7 2016.

[67] P. Malhotra, V. TV, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder," 8 2016.

[68] D. Wu, C. Jennings, J. Terpenny, R. X. Gao, and S. Kumara, "A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests," *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 139, 7 2017.

[69] V. Mathew, T. Toby, V. Singh, B. Maheswara Rao, and M. Goutham Kumar, "Prediction of Remaining Useful Lifetime (RUL) of Turbofan Engine using Machine Learning," *Proceedings of 2017 IEEE International Conference on Circuits and Systems (ICCS 201*, 2017.

[70] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," 5 2015.

[71] F. O. Heimes, "2008 INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT Recurrent Neural Networks for Remaining Useful Life Estimation," tech. rep., 2008.

[72] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries," *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 5695–5705, 7 2018.

[73] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, *Long Short Term Memory Networks forAnomaly Detection in Time Series*. Ciaco, 2015.

[74] A. Ray and S. Tangirala, "Stochastic Modeling of Fatigue Crack Dynamics for On-Line Failure Prognostics," Tech. Rep. 4, 1996.

[75] A. Dasgupta and M. Pecht, "Material Failure Mechanisms and Damage Models," Tech. Rep. 5, 1991.

[76] Y. Hu, P. Baraldi, F. Di Maio, and E. Zio, "Online Performance Assessment Method for a Model-Based Prognostic Approach," *IEEE Transactions on Reliability*, vol. 65, pp. 718–735, 6 2016.

[77] E. Ramasso and A. Saxena, "Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset," *PHM 2014 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014*, pp. 612–622, 2014.

[78] C. Zheng, W. Liu, B. Chen, D. Gao, Y. Cheng, Y. Yang, X. Zhang, S. Li, Z. Huang, and J. Peng, "A Data-driven Approach for Remaining Useful Life Prediction of Aircraft Engines," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-Novem, pp. 184–189, 2018.

[79] T. Wang, J. Yu, D. Siegel, and J. Lee, "A similarity-based prognostics approach for remaining useful life estimation of engineered systems," *2008 International Conference on Prognostics and Health Management, PHM 2008*, no. September 2014, 2008.

[80] G. Sateesh Babu, P. Zhao, and X.-L. Li, "Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life," in *Database Systems for Advanced Applications* (S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, and H. Xiong, eds.), (Cham), pp. 214–228, Springer International Publishing, 2016.

[81] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "HYBRID DEEP NEURAL NETWORK MODEL FOR REMAINING USEFUL LIFE ESTIMATION Electrical and Computer Engineering , Concordia University , Montreal , QC , Canada H3J 1P8 Concordia Institute for Information System Engineering (CIISE), Concordia University , Cana," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3872–3876, 2019.

[82] L. Jayasinghe, T. Samarasinghe, C. Yuenv, J. C. Ni Low, and S. Sam Ge, "Temporal convolutional memory networks for remaining useful life estimation of industrial machinery," *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2019-Febru, pp. 915–920, 2019.

[83] S. Rusell and P. Norvig, "Artificial Intelligence A Modern Approach Third Edition," tech. rep., Prentice Hall, 2010.

[84] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," 10 2017.

[85] O. Vincent and O. Folorunso, “A Descriptive Algorithm for Sobel Image Edge Detection,” *Proceedings of the 2009 InSITE Conference*, 2009.

[86] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, “Remaining useful life estimation of engineered systems using vanilla LSTM neural networks,” *Neurocomputing*, vol. 275, pp. 167–179, 1 2018.

[87] S. Ruder, “An overview of gradient descent optimization algorithms,” pp. 1–14, 2016.