

N° d'ordre: /Faculté/UMBB/2020

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University of M'hamed Bougara
Faculty of Hydrocarbons & Chemistry



Department of Hydrocarbons Transportation & Equipment

Branch: Hydrocarbons

Major: Mechanical Engineering

Option: Upstream Mechanical Engineering (MACP-15)

Master Thesis

Presented by: LATRACH Abdeldjalil

Title:

Application of Deep Learning for
Predictive Maintenance of
Oilfield Equipment

In front of the jury:

HALIMI Djamel

Docteur

UMBB

Adviser



DEDICATION

*I dedicate this humble work to my dear parents,
to everyone who stood by my side and supported me through my academic journey,
to everyone who ever taught me a letter.*

ACKNOWLEDGEMENT

I would like to thank EXPRO family for the host during my internship period. I acquired during those few weeks of my stay there—which was suddenly and sadly terminated due to COVID-19 lockdown—more valuable knowledge than what I was taught during whole semesters of university. A special thank for Mr. Aziz Menai and Mr. Zohir Meksen and all family of production department.

I would like to thank the whole opensource software community for providing amazing set of tools and voluntary technical support which without it this work wouldn't have been brought out in this form.

I would like to thank the whole family of Petroleum Club at University of Boumerdes. These past few years were full of memories and valuable experiences which aren't and couldn't be taught in a classroom.

Abstract

This thesis explored applications of the new emerging techniques of artificial intelligence and deep learning (neural networks in particular) for predictive maintenance, diagnostics and prognostics. Many neural architectures such as fully-connected, convolutional and recurrent neural networks were developed and tested on public datasets such as NASA C-MAPSS, Case Western Reserve University Bearings and FEMTO Bearings datasets to diagnose equipment health state and/or predict the remaining useful life (RUL) before breakdown. Many data processing and feature extraction procedures were used in combination with deep learning techniques such as dimensionality reduction (Principal Component Analysis) and signal processing (Fourier and Wavelet analyses) in order to create more meaningful and robust features to use as an input for neural networks architectures. This thesis also explored the potential use of these techniques in predictive maintenance within oil rigs for monitoring oilfield critical equipment in order to reduce unpredicted downtime and maintenance costs.

Keywords: predictive maintenance, prognostics, deep learning, neural networks, signal processing

Contents

Introduction	1
1 Towards a Data-Driven Prognostics Approach	4
1.1 Prognostics of mechanical equipment	4
1.2 Estimating Remaining Useful Life	5
1.3 Physical, Data-Driven and Hybrid Approaches	5
1.3.1 Physics-based models	5
1.3.2 Data-Driven Models	6
1.3.3 Statistical Models	6
1.3.4 Machine Learning	6
1.3.5 Hybrid Models	7
1.4 Why a Data-Driven Approach?	7
1.5 Conclusion	8
2 Steps of a Data-Driven Approach	9
2.1 Data acquisition	9
2.2 Features Extraction	10
2.2.1 Signal analysis	10
2.2.2 Temporal Analysis	11
2.2.3 Frequency-Domain Analysis	11
2.2.4 Time–Frequency Analysis	11
2.2.5 Fourier analysis	12
2.2.6 Wavelet transform	12
2.2.7 Dimensionality reduction	15
2.3 Diagnostics	16

2.4	Prognostics	16
2.5	Maintenance decision	17
2.6	Conclusion	17
2.7	Conclusion	17
3	Introduction to Artificial Neural Networks	18
3.1	Structure of artificial neural networks	18
3.2	Feedforwad networks: from input to output	19
3.3	Activation functions	19
3.4	Network training	20
3.4.1	Cost function	20
3.4.2	Gradient Descent	21
3.4.3	Convex and non-convex functions	21
3.4.4	Global and local minima	22
3.4.5	Saddle points	23
3.4.6	Optimization of multilayer networks	23
3.4.7	Backpropagation	24
3.5	Recurrent Neural Networks	25
3.5.1	Long Short-Term Memory	25
3.6	Convolutional neural networks	27
3.7	Conclusion	27
4	Equipment Health Assessment using Artificial Neural Networks	28
4.1	Introduction to NASA C-MAPSS dataset	28
4.2	Visualization of equipment degradation	29
4.3	Engines health classification	31
4.4	Remaining Useful Life prediction	33
4.4.1	Remaining Useful Life modeling	33
4.4.2	RUL prediction using feed-forward network	33
4.4.3	Improving RUL prediction using LSTM networks	35
4.5	Application to oilfield equipment	37
4.6	Conclusion	38
5	Bearings Faults Diagnostics and Prognostics	41
5.1	Case Western Reserve University Bearings Data	41

5.2	Generating data from raw vibration signals	42
5.3	Case study: Bearings faults diagnostics using neural networks	43
5.3.1	Network architecture	43
5.3.2	Training process	43
5.3.3	Network results discussion	44
5.4	Case study: Bearings faults prognostics using neural networks	45
5.4.1	Introduction to FEMTO database	46
5.4.2	Generating data from scaleograms	47
5.4.3	Detecting bearings failure using convolutional neural networks .	49
5.4.4	The need for new prognostics features	51
5.4.5	Trigonometric features and cumulative descriptors	52
5.5	Application to oilfield equipment	55
5.5.1	Top Drive	55
5.5.2	Top Drive components	56
5.5.3	Proposed approach for Top Drive monitoring using neural networks	60
5.6	Conclusion	61
	Conclusion	62

List of Figures

1.1	Classification of prognostics approaches	5
1.2	Taxonomy of prognostics approaches [1]	6
1.3	Series hybrid configuration (Figure adapted from [2])	7
1.4	Parallel hybrid configuration (Figure adapted from [2])	7
2.1	Different signal processing techniques	11
2.2	Signal in the time domain and its fast Fourier transform	12
2.3	Time—frequency resolution plane	13
2.4	Different types of wavelets	13
2.5	Scaleogram of vibration data snapshot	14
2.6	Discrete wavelet transform (DWT) as a filter bank	15
2.7	Level 3 signal decomposition using DWT	15
3.1	Structure of an artificial neural network	19
3.2	Different activation functions	20
3.3	Condition of convexity	22
3.4	Example of convex and non-convex functions	22
3.5	Non-convex function with global and local minima	23
3.6	Saddle point	23
3.7	Forward (left) and Backward (right) passes	24
3.8	Recurrent neural networks architecture	25
3.9	Long Short-Term Memory Cell	26
3.10	Applying convolutions (by multiplying the input by a kernel) iteratively to different regions (dark blue) of the 2D input (whole blue square). Each iteration gives a numerical value (dark green). The result of all iterations is the green 2D output [3].	27

4.1	Simplified diagram of engine simulated in C-MAPSS [4]	28
4.2	Development of 6 sensors outputs from one of the engines (normalized)	30
4.3	Equipment health degradation (lighter colors indicate advancing of health degradation) of four turbofan engines from C-MAPSS dataset	30
4.4	Training process of C-MAPSS classifier	32
4.5	C-MAPSS classifier ROC curve on test set	32
4.6	Different RUL modeling choices	33
4.7	Training process of RUL predictor	34
4.8	Prediction results on two test units	35
4.9	LSTM training process	36
4.10	LSTM prediction results on two test units	37
4.11	List of sensors installed on oil rigs for monitoring	39
5.1	Bearing's components	41
5.2	Data generation by converting the signal into 64×64 images	42
5.3	Converted signals of different faults types	43
5.4	Classifier training	45
5.5	Confusion matrix of bearings faults classification using CNN	46
5.6	PRONOSTIA platform [5]	47
5.7	Scaleograms of different stages of Bearing1_1 life (horizontal vibrations)	48
5.8	Scaleograms of different stages of Bearing1_1 life (vertical vibrations) .	49
5.9	Bearings health state classifier training	50
5.10	Bearings health state classifier ROC curve on test set	51
5.11	Trigonometric features of Bearing1_1	53
5.12	Classic and trigonometric cumulative features	54
5.13	Features fitness	55
5.14	Topdrive Bentec 500-HT	56
5.15	Bentec 500-HT Top Drive sub-assemblies	57
5.16	Bentec 500-HT Top Drive drilling unit	58
5.17	Main components of the drive	59
5.18	SKF tapered roller thrust bearing [6]	59
5.19	Bearing position within the Top Drive [7]	60
5.20	Proposed additional vibration sensors to monitor the drive	61

List of Tables

3.1	Mathematical definition of several activation functions	20
4.1	C-MAPSS outputs to measure system response.	29
4.2	Units number and cycles length statistics in C-MAPSS data	29
4.3	C-MAPSS classifier architecture	31
4.4	C-MAPSS classifier metrics on test set	33
4.5	Fully-connected network architecture for RUL prediction	34
4.6	LSTM network architecture for RUL prediction	36
4.7	LSTM training results	37
4.8	Honeywell oil rig sensors list	40
5.1	CWRU bearings dimensions and faults frequencies	42
5.2	44
5.3	CNN architecture	44
5.4	CNN training results	44
5.5	Different operating conditions and corresponding bearings in FEMTO dataset	48
5.6	Bearings health state classifier architecture	50
5.7	Training result	51
5.8	Additional metrics for network performance	51
5.9	Prognostics trigonometric and classic features [8]	53
5.10	Monotonicity difference between trigonometric and classic features and their cumulative descriptors	54
5.11	Trendability difference between trigonometric and classic features and their cumulative descriptors	54

Acronyms

AUC Area Under Curve.

CBM Condition-Based Maintenance.

CM Condition Monitoring.

CNN Convolutional Neural Networks.

CWT Continuous Wavelet Transform.

DD Data-Driven.

LCM Life Cycle Management.

LSTM Long Short-Term Memory.

PHM Prognostics and Health Management.

PM Preventive Maintenance.

RNN Recurrent Neural Networks.

ROC Receiver Operating Characteristics.

RUL Remaining Useful Life.

“All models are wrong, but some
are useful”.

George E. P. Box

Introduction

All machines in general, and mechanical machines in particular, are subject to degradation of their condition and deterioration of their performance over time, possibly leading to their failure. These failures can have negative impacts on economic, human, environmental aspects. Machine failure is an intrinsic property of these systems (due to their inherent physical properties), it can be - in order to avoid negative results - partially prevented, delayed and even foreseen, but it can never be totally prevented or stopped, this is mainly achieved through maintenance.

Maintenance is defined as a set of activities intended to maintain or restore a utility unit in a state in which it can perform a required function [9].

The way in which industrial maintenance is performed has evolved with advances in technology from the beginning of the industrial revolution to the present day. Its most basic form is the unplanned corrective maintenance that is carried out after a failure has occurred, in order to restore a functional unit to a state where it can perform a required function[9]. Prior to World War II, machinery was relatively simple and production demand was moderate, so that it could be maintained after a breakdown. After the war and with the rebuilding of the industry, the market became more competitive and less tolerant of downtime, so the industry turned to preventive maintenance (Preventive Maintenance: PM) which is carried out at predetermined intervals or according to prescribed criteria to reduce the probability of failure of a functional unit [9]. Corrective and preventive maintenance are both obsolete. The goal is to replace "post-mortem" and "blind" maintenance with "just-in-time" maintenance. It was therefore only a matter of time before the idea of predictive or condition-based maintenance (Condition-Based Maintenance: CBM) emerged, which differs from other forms by basing the need for intervention on the actual state of the machine rather than on a pre-set schedule[10]. There is disagreement in the literature on this classification (of CBM as preventive maintenance or a separate form)[11]. The difference in taxonomy is not of real interest to the current discussion.

A new field that has recently emerged is that of health prognosis and management (Prognostics and Health Management: PHM). PHM has emerged as an essential approach to achieving competitive advantages in the global marketplace by improving reliability, maintainability, safety and affordability. Like CBM, PHM is a discipline that has emerged in industry with a modest presence in the academic field, particularly in military applications[12]. (e.g. the development of the F-35 fighter plane [13]). The concepts and components of the PHM have been developed separately in

many fields such as mechanical, electrical and statistical engineering, under various names[14]. While CBM focuses on system monitoring, PHM is a more integrated approach that aims to provide guidelines for system health management. It is therefore a life cycle management philosophy (Life Cycle Management: LCM) that emphasizes predictability (i.e., prognosis) of failures and maintenance. This is usually achieved through the adoption of a monitoring strategy, which can be a technique of CBM [12]. The main goal of the prognosis (also CBM) is the estimation of the remaining useful life (Remaining Useful Life: RUL) given the current state of the equipment and its historic operational profile[15].

The PHM operates at a slightly higher level than the CBM, since it has the clear ambition to enable health management. This is an activity related to LCM, which means that an approach is followed to optimize all (maintenance) activities during the complete life-cycle of the equipment. This includes choosing an appropriate maintenance policy, defining the intervention schedule and deciding when a piece of equipment should be taken out of service. CBM does not offer this extensive LCM support. The PHM domain prescribes neither a specific maintenance concept nor a monitoring strategy. However, in typical PHM studies, CBM or other maintenance policies are adopted and in many cases, condition monitoring techniques (Condition Monitoring: CM) are applied[12].

As mentioned earlier, all these efforts to develop maintenance strategies have been mainly due to the high costs of production loss and reactive maintenance in capital-intensive industries such as Onshore and Offshore projects in the oil and gas industry are mainly investments, which can have serious financial and environmental consequences if a catastrophic failure occurs. Therefore, an effective approach to maintenance management is essential to continue production safely and reliably [16]. Offshore organizations experience an average financial impact of \$38 million per year due to unplanned downtime (for the worst performing organizations, \$88 million). Less than 24% of operators describe their maintenance approach as predictive and based on data and analysis. More than three-quarters take a reactive or time-based approach. Operators using a predictive data-based approach experience on average 36% less unplanned downtime than those using a reactive approach. This can translate, on average, into a \$34 million per year decline in bottom line earnings[17].

In this thesis, a data-driven prognostic approach will be introduced, outlining the various steps required from data acquisition to the estimation of RUL. Due to the commercial and technical sensitivity of the data in the petroleum field (and all industries with high added value) and therefore the impossibility to obtain them, the public databases (NASA Ames Data Repository [?]) that are usually used to evaluate prognostic algorithms in the literature will be used here as well. An approach will be proposed to transfer the knowledge obtained to concrete applications in the petroleum industry.

This thesis is divided as follows:

Chapter 01 Towards a Data-Driven Prognostics Approach

Chapter 02 Steps of a Data-Driven Approach

Chapter 03 Introduction to Artificial Neural Networks

Introduction

Chapter 04 Equipment Health Assessment using Artificial Neural Networks

Chapter 05 Bearings Faults Diagnostics and Prognostics

Conclusion

Chapter 1

Towards a Data-Driven Prognostics Approach

The aim of this chapter is to present the different prognostic approaches with a detailed taxonomy, the different steps of any prognostic approach will be described and then the emphasis will shift towards data-driven methods.

1.1 Prognostics of mechanical equipment

Some complex systems, especially in the oil industry, operate under very severe conditions (offshore, desert, etc.), which can lead to the occurrence of failures and their degradation. Breakdowns and unplanned shutdowns inevitably cause production losses, which can have enormous economic consequences. With these economic constraints, maintenance programs must be developed to minimize the probability of failures, thereby reducing the cost. As discussed in the introduction to this thesis, these programs must be based on the principles of Condition-Based Maintenance and Prognostics and Health Management.

Prognostics and health management (Prognostics and Health Management) has two major aspects [18]:

1. **Prognostics:** Predictive diagnostics, which includes determining the remaining useful life (service life) of a component or a full system.
2. **Health Management:** the ability to make decisions regarding maintenance actions based on diagnostic/prognostic information, available resources and operational demand.

1.2 Estimating Remaining Useful Life

The main objective of the prognostics is to estimate the remaining useful life (Remaining Useful Life) of the system. RUL is defined according to the equation 1.1:

$$RUL = t_f - t_c \quad (1.1)$$

Where t_f is the predicted time for the occurrence of the failure and t_c is the current time (the time when the prediction is made).

1.3 Physical, Data-Driven and Hybrid Approaches

Any prognostic approach can be based on physical models, data-driven models or a hybrid combination of the two (Figure 1.1).

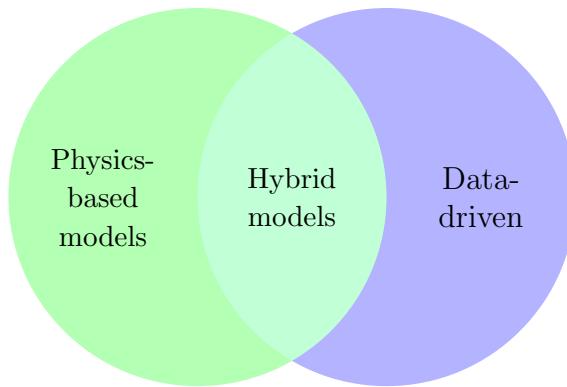


Figure 1.1: Classification of prognostics approaches

These three categories constitute a general classification based on the approach followed, each of which can be subdivided into sub-categories. A detailed taxonomy is presented in Figure 1.2.

1.3.1 Physics-based models

Physics-based models assess the health of the system using an explicit mathematical formulation (white boxes) developed on the basis of a scientific and technical understanding of its behavior. However, the main advantage of these physical models is the use of degradation models to predict long-term behavior [19]. Physical approaches are capable of providing an accurate estimate of the health of the system if the physical model is developed with a complete understanding of the failure mechanisms and efficient estimation of the model parameters. However, for some complex mechanical systems, it is difficult to understand the physics of damage, which limits the application of these approaches [20].

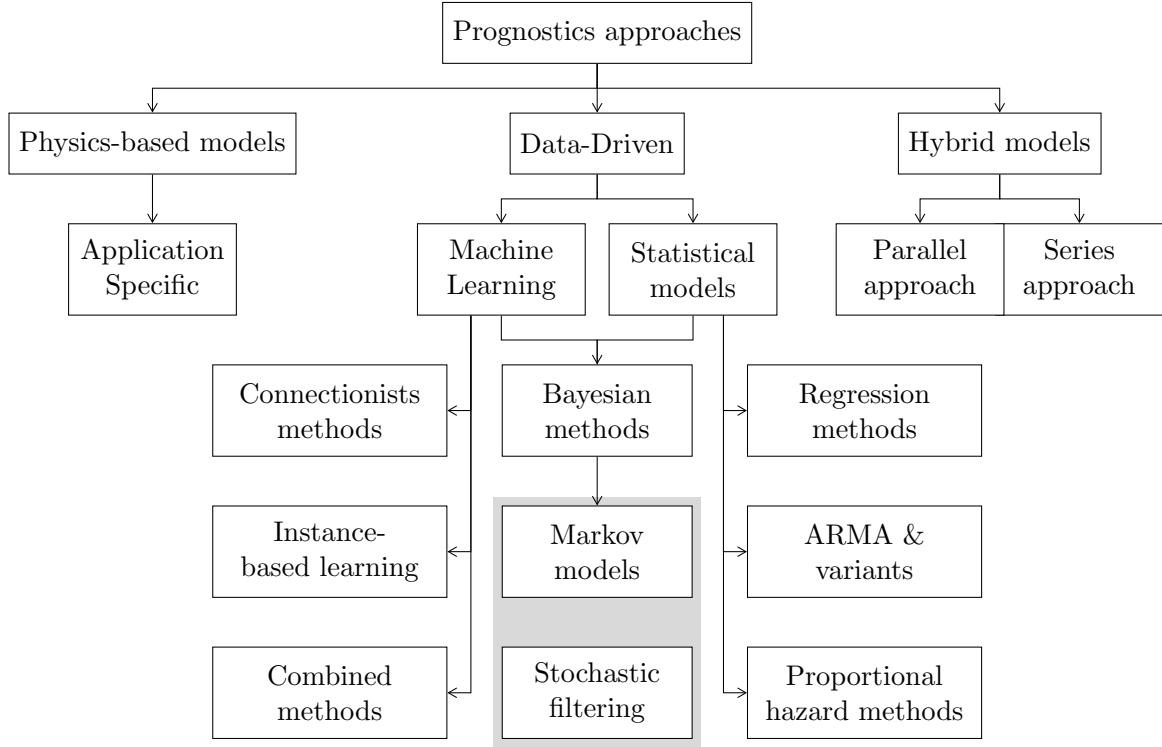


Figure 1.2: Taxonomy of prognostics approaches [1]

1.3.2 Data-Driven Models

Data-Driven models rely on previously collected data (monitoring data, data on operational parameters, ...) to establish a model capable of assessing the health of the system and predicting its behavior and degradation. Contrary to physical models, and as their name indicates, Data-Driven models do not rely on human knowledge but mainly on historical data collected to model the degradation process. Usually, they are considered as black boxes.

1.3.3 Statistical Models

The statistical approach is based on the construction and fitting of a probabilistic model using historical data without depending on any physical or technical principles. Si et al. [21] presented a review of statistical approaches. According to this review, many models fall into this category such as regression models (e.g. linear regression), auto-regressive moving average and its variants, stochastic filtering techniques (e.g. Kalman filter, particles filter, ...).

1.3.4 Machine Learning

Machine Learning is a field of Artificial Intelligence that has exploded in popularity during recent years and has made breakthroughs in many areas such as computer vision and natural language processing. Machine learning models are black box models that allow even very complex mappings from input to output to be discovered. Many types

of algorithms fall into this category such as connectionist methods (e.g. artificial neural networks), context learning (e.g. support-vector machines). Different approaches can be combined together to create mixed models that may perform better than a single model.

1.3.5 Hybrid Models

Hybrid models are a combination of a physical model and an Data-Driven model. There are two types of hybrid models depending on how the two types of models are combined. The Data-Driven model can be integrated into a physical model in a series configuration (Figure 1.3) where it is used to adjust the parameters of the physical model which is then used to make predictions.

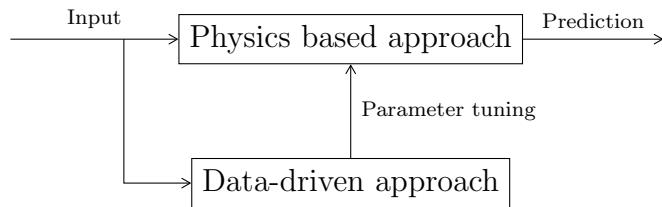


Figure 1.3: Series hybrid configuration (Figure adapted from [2])

The two types of models can be combined in a parallel configuration (Figure 1.4) where the two models make separate predictions that can be combined to obtain the final estimate.

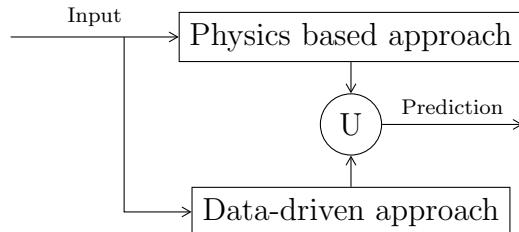


Figure 1.4: Parallel hybrid configuration (Figure adapted from [2])

1.4 Why a Data-Driven Approach?

As mentioned previously, understanding the degradation process for very complex systems is extremely difficult, which is why the development of physical models is very problematic for these systems.

The last 20 years have seen great progress in the development of new detection techniques, prognostics/diagnostics methods and in the application of computer-aided analysis methods.

It is interesting to note that at the 2002 workshop on condition-based maintenance organized by the Advanced Technology Program of the United States National Insti-

tute of Standards and Technology (NIST), the following obstacles to the widespread application of CBMs were identified:

- The impossibility of accurately and reliably predicting the remaining service life of a machine.
- The inability to continuously monitor a machine.
- The Inability of maintenance systems to learn and identify impending failures and recommend action.

These barriers can be redefined as deficiencies in prognosis, detection and reasoning. These and other limitations in the current implementation of condition-based maintenance techniques have, of course, been recognized by others and have led to the development of programs (e.g. in the military field) to overcome them [18].

Today, sensors in industry have become inexpensive and ubiquitous, computing power has increased exponentially which has allowed the development of more advanced algorithms and computing tools: the artificial intelligence revolution and machine learning. The industry generates huge amounts of data in many areas, the majority of which is untapped. Exploiting this data using the latest technological advances can increase profits, drastically reduce costs and prove to be a huge economic advantage.

1.5 Conclusion

Adoption of data-driven prognostics models can be helpful especially when behavior of degradation process is ambiguous and developing physics-based models to quantify the health state of complex systems is complicated and their results become unreliable. Recent developments in sensoring techniques, increase in available computation power (i.e. faster and cheaper processing units) and also abundance of—unexploited—monitoring data has provided the needed framework for adopting these data-driven models which are easier to develop, deploy and automate than their physics-based counterparts.

Chapter 2

Steps of a Data-Driven Approach

A Data-Driven prognostics approach (an approach that uses system's operation data, like historic monitoring data, to build a model that is used to predict remaining useful life) must go through multiple stages—from data acquisition to estimating remaining service life. In this chapter, these different steps will be discussed in detail.

2.1 Data acquisition

A signal is a function that conveys information about the behavior of a system or the attributes of a phenomenon. Signals occur naturally and are also synthesized. A signal is not necessarily an electrical quantity. However, to perform activities such as synthesizing, transporting, recording, analyzing and modifying signals, it is often convenient to use a signal in the form of an electrical quantity [22].

Data acquisition is a process of capturing and storing different types of monitoring data (i.e. signals) from various sensors installed on the monitored equipment. It is the first machine prognostics process, which provides basic condition monitoring information for subsequent processes. A data acquisition system is composed of sensors, data transmission devices and data storage devices [20]. Condition monitoring data is very versatile. It can be vibration data, acoustic data, oil analysis data, temperature, pressure, humidity, weather or environmental data, and so on. Different sensors, such as micro-sensors, ultrasonic sensors and acoustic emission sensors have been designed to collect different types of data. Wireless technologies, such as Bluetooth, have provided a cost-effective alternative to data communication [15].

Although research on advanced concepts such as wireless sensor networks and energy harvesting to power autonomous sensors is ongoing, data acquisition (sensors) and manipulation are now fairly well established. Consequently, much of the research

in this discipline focuses on analyzing the data obtained to extract information [12]. Because this discipline is well developed, many new data acquisition facilities and techniques have been designed and applied in modern industries. These powerful and versatile facilities have made data acquisition for PHM implementation more practical and feasible[23].

2.2 Features Extraction

The data-driven prognostics approach is mainly used when it is difficult to understand the physical behavior of a complex system. Understanding the behavior and interaction of the different elements that lead to machine degradation is the starting point for developing a physical model for prognostics. On the other hand, this approach uses condition monitoring data to implicitly model its behavior. The models used in the data-driven approach use monitoring data to model complex behavior and capture complex models, but they are considered as black boxes: they do not necessarily provide insight into the process. In general, the performance of these models depends on the quality of the input (i.e. data). The human part cannot perform the task that the model performs, but processing the input data can significantly increase the results. This processing is necessary because the sensor data on which the model is based is usually redundant, noisy and incomplete, and there are many reasons for these imperfections.

Feature extraction is an important pre-processing step in the development process of machine learning models and directly influences the performance of the model. Therefore, this step must be performed carefully in order to extract meaningful features from the raw data. Vibration data contains very useful information about the state of the system, but requires extensive pre-processing before it can be used as input data for a specific model. This chapter describes some of the signal processing techniques used in traditional vibration analysis, but in this context they will be used as feature extractors for a neural network architecture.

2.2.1 Signal analysis

Signal processing is the study and analysis of stored signals to reveal their properties-which may not be apparent at first glance-using a set of algorithms and techniques. In the context of condition monitoring, these properties revealed by signal processing may be indicative of the health of the machine. Signal processing is a well-established and mature sub-domain of electrical engineering, with many techniques and algorithms proposed in the literature.

Signal processing can be divided into three categories: **Time-Domain Analysis**, **Frequency-Domain Analysis** and **Time-Frequency Analysis**.

2.2.2 Temporal Analysis

. The original measurements of signals that are usually repeatedly sampled between predefined time intervals are in the form of a time series. Thus, time domain analysis is directly based on the original measurement [23].

2.2.3 Frequency-Domain Analysis

Frequency domain analysis is based on signals transformed in the frequency domain. The advantage of frequency-domain analysis over time-domain analysis is its ability to decompose the original signals into a series of frequency components. The most commonly used frequency domain analysis is spectrum analysis using Fast Fourier Transform (FFT). The main idea of spectrum analysis is to isolate and locate certain frequency components of interest related to the fault characteristics of [24] machines.

2.2.4 Time–Frequency Analysis

The problem with time-domain analysis and frequency-domain analysis is that each has no information on the other (time-domain analysis has no information on the frequency-domain, and frequency-domain analysis has no information on the time-position).

Thus, time–frequency domain analysis, which studies measurement signals in the time and frequency domains, has been applied to the analysis of non-stationary measurement signals. Time–frequency analysis describes the characteristics of measurement signals in two-dimensional functions of time and frequency to better reveal machine failure modes.

Figure 2.1 presents different techniques for each type of analysis:

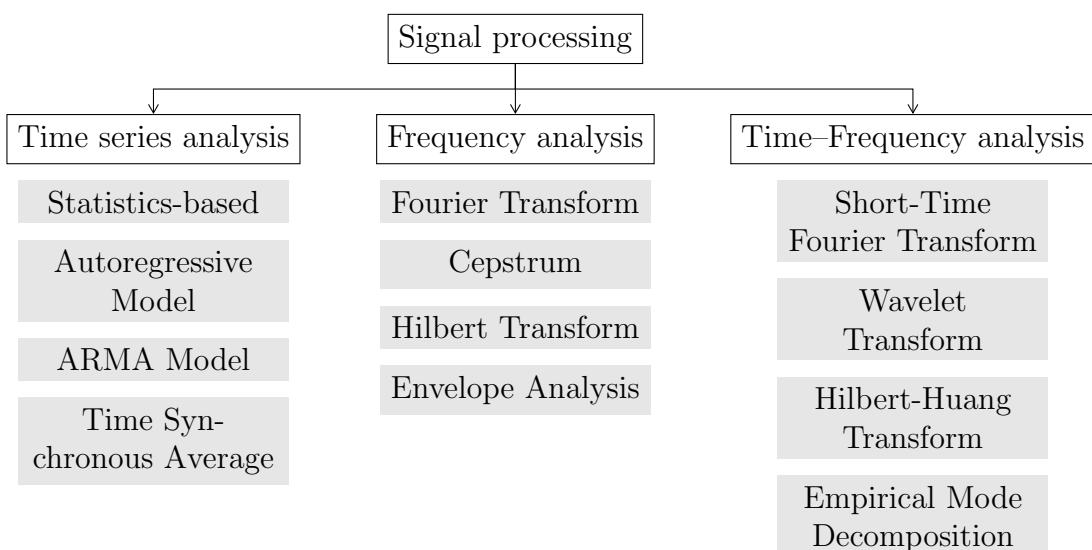


Figure 2.1: Different signal processing techniques

2.2.5 Fourier analysis

Fourier analysis, also called harmonic analysis, of a periodic signal $x(t)$ is the decomposition of the series into summation of sinusoidal components, where each sinusoid has a specific amplitude and phase.

The Fourier transform (FT) of a signal $x(t)$ can be mathematically given by equation 2.1:

$$X(w) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (2.1)$$

In practical applications of digital signal processing where signals are discrete in time rather than continuous (e.g. vibration analysis) a discretized version called discrete Fourier transform (DFT) is used instead, it is expressed mathematically by equation 2.2:

$$X(w) = \sum_{-\infty}^{\infty} x(n)e^{-j\omega n} \quad (2.2)$$

Fast Fourier transform (FFT) is an effective algorithm used to implement DFT in computers. Figure 2.2 shows a signal in its waveform (or time domain) and its corresponding spectrum (frequency domain) obtained using FFT algorithm. The spectrum shows the frequency components present in the signal:

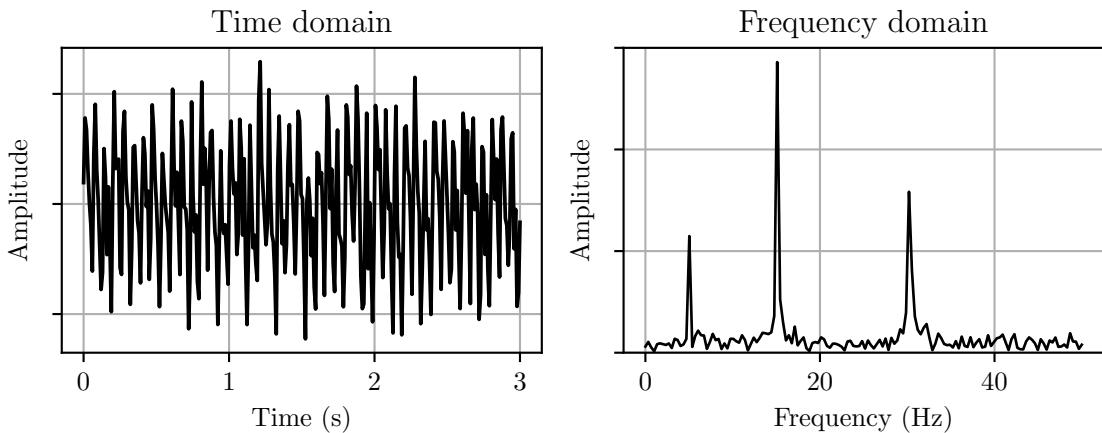


Figure 2.2: Signal in the time domain and its fast Fourier transform

2.2.6 Wavelet transform

Wavelet transform is also a spectral analysis tool, like Fourier transform. The main difference is that Fourier transform decomposes the signal into sinusoidal components, but wavelet transform decomposes it into a set of oscillatory functions called **wavelets**. Unlike sinusoids, wavelets are localized in time, thus wavelet transform doesn't only provide information about the frequency present in a signal but also the time of their

occurrence. Wavelet transform is a much better solution than Fourier transform when studying non-linear non-stationary signals (i.e. its frequency components vary with time).

Figure 2.3 shows the difference in time and frequency resolutions between different methods. In the waveform, the signal has absolute resolution in time and zero resolution in frequency. Fourier transform on the contrary transforms the signal totally into the frequency domain, therefore it has absolute resolution in frequency but no resolution in time. Short-time Fourier transform is calculated identically to Fourier transform but it is performed on separate segments of the original signal to preserve some resolution in time. Wavelet transform on the other hand exhibits a high time resolution for high frequencies and high frequency resolution for low frequencies:

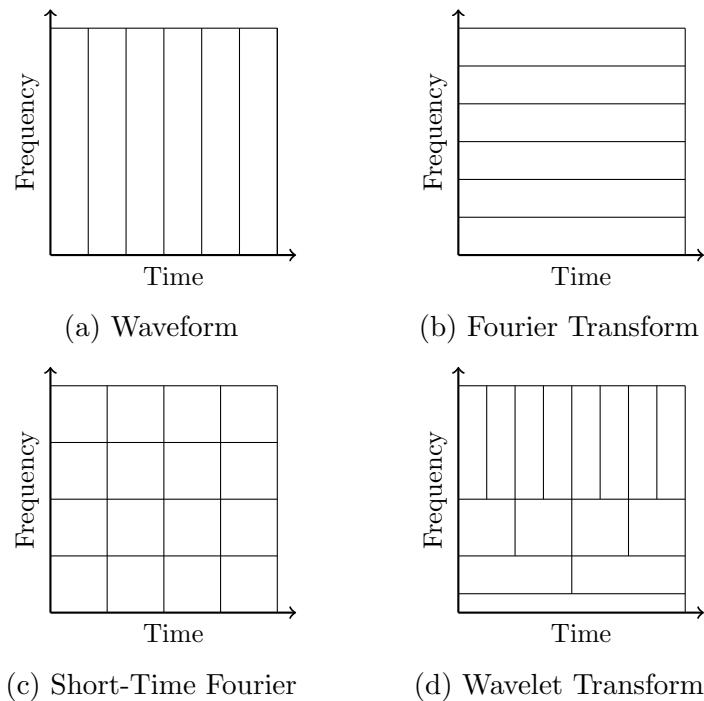


Figure 2.3: Time—frequency resolution plane

There are a wide variety of wavelets that serve different purposes like Morlet wavelet, Daubechies wavelet and many others.

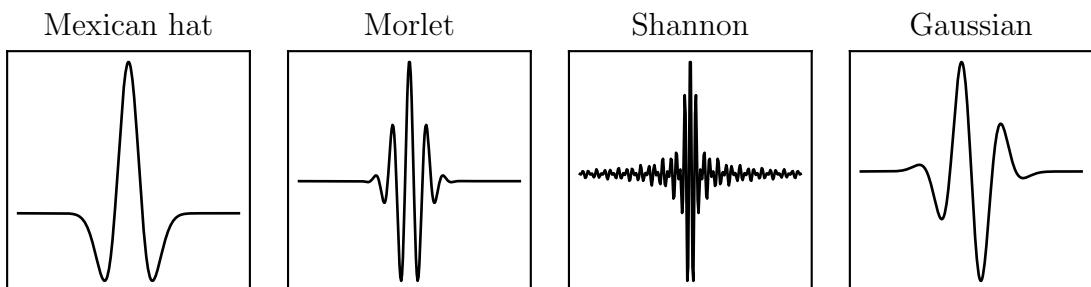


Figure 2.4: Different types of wavelets

Continuous wavelet transform

Mathematically, continuous wavelet transform is defined by equation 2.3:

$$CWT_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t - \tau}{s} \right) dt \quad (2.3)$$

Where $x(t)$ is the original signal, ψ^* is a function called the **mother wavelet**; s and τ are the **scale** and **translation** parameters respectively. The original signal is multiplied by the mother wavelet which is scaled using different scales then translated over the signal.

The output of CWT is a scaleogram like the one in figure 2.5 which is a scaleogram (filled contour plot) of vibrations data snapshot of 25ms:

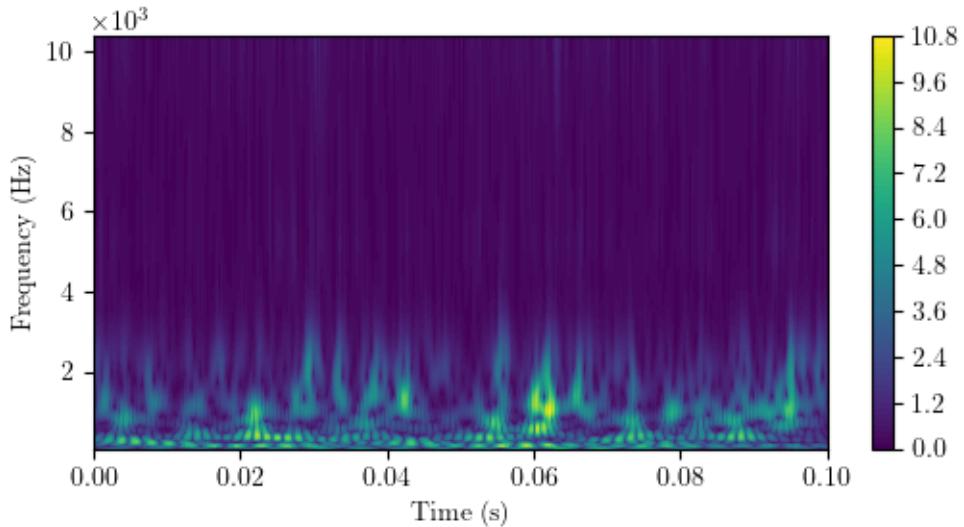


Figure 2.5: Scaleogram of vibration data snapshot

The x and y axes represent time and frequency respectively. Different colors indicate the power (i.e. amplitude) of each frequency (y-axis) during each instant of time (x-axis) which—unlike Fourier transform—provides information about the frequencies present in the signal and also the instances of time when these frequencies are present.

Discrete wavelet transform

In practical applications, discrete wavelet transform (DWT) is implemented as a filter bank where the signal is passed through low- and high-pass filters to obtain **approximation** and **decomposition coefficients**. Figure 2.6 shows a DWT with 2 levels of decomposition which yields 2nd order approximation and decomposition coefficients:

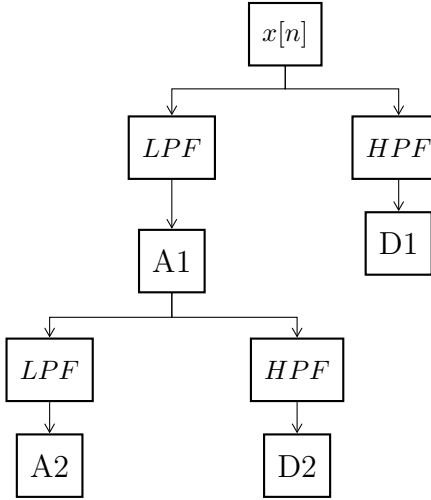


Figure 2.6: Discrete wavelet transform (DWT) as a filter bank

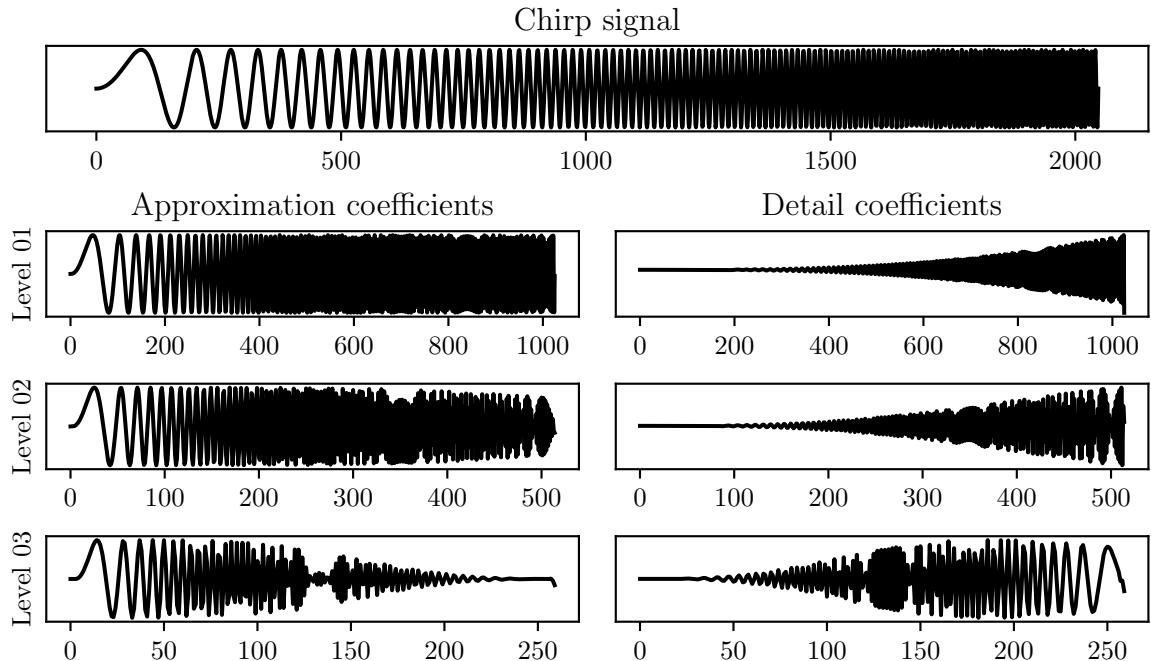


Figure 2.7: Level 3 signal decomposition using DWT

DWT returns two sets of coefficients: **approximation coefficients** associated with the low pass filter and **detail coefficients** associated with the high pass filter of the DWT. By applying DWT again on the approximation coefficients the next level of decomposition can be obtained. At each level the original signal is downsampled by a factor of 2, this fact imposes a limitation on the possible number of decomposition levels for a given signal.

2.2.7 Dimensionality reduction

Dimensionality reduction refers to the process of taking high-dimensional data and finding a good representation of this data in a lower dimension while preserving its

original characteristics. Dimensionality reduction is performed for feature extraction by finding the main components or for visualization (by reducing the number of dimensions to 2 or 3). There are many dimensionality reduction techniques and algorithms such as:

- Principal Component Analysis (PCA)
- Autoencoders
- t-Distributed Stochastic Neighbor Embedding (t-SNE)

It should be noted that for prognoses, where the data used consist of sensor inputs where each variable has a physical meaning and direct interpretation, performing a dimensionality reduction will give the main components, but the new variables *will lose their physical interpretability* and *shall be treated as abstractions*.

2.3 Diagnostics

Diagnostics is a process of identifying and determining the relationship between the information obtained in the measurement space and the failure modes of the machines in the failure space. Diagnostics consists of three main steps: fault detection, fault isolation and fault identification. Fault detection is the task of indicating whether a fault has already occurred in the monitored machines. Fault isolation consists of finding the component of the fault and the position of the fault. Fault identification is the final step in diagnosis, which attempts to determine the mode and severity of the failure. The three steps are interrelated. This last step is based on the results of the first one and therefore cannot be performed individually [25].

2.4 Prognostics

Diagnosis is the analysis of the later event and prognosis is the analysis of the earlier event. Prognosis is much more effective than diagnosis in achieving performance without stopping production. However, diagnosis is necessary when the prediction of prognostic errors fails and an error occurs [15]. Prognostics are the forecast or prediction of the future performance of a system, this prediction is based on its current state. The prediction is performed using a model, types of prognostic models are already discussed in details in section 1.3 but the focus of this current discussion is geared towards data-driven models. More concretely, what these models predict is the remaining useful life (RUL) defined in section 1.2. Prognostics models—by estimating RUL—aim at scheduling maintenance actions according to the predictions and production constraints associated with the machine in order to achieve zero unscheduled equipment downtime.

2.5 Maintenance decision

The next and the last step of any prognostics approach is to use the output of the developed model (i.e. predictions, or estimations for the system's RUL) in making better and more accurate preventive maintenance actions. Maintenance decision makers should analyze the outputs of the model prior to taking any actions, since any model has an inherent error associated with its predictions. Usually, data-driven models provide confidence intervals associated with their predictions which must be considered carefully. Also it must be noted that those confidence intervals are reflection of the models certainty of its predictions based on the data provided for the training, in real applications a new unforeseen failure mode can occur which may not have happened before, thus it wasn't provided for the model as part of the data used to construct it. This is more or less associated with the ability of different types of models to generalize for new unforeseen data, but in general it is well-known fact that any data-driven model will be less reliable when performing predictions using data that the model didn't see before [26]. That's why providing quality data that reflects different degradation patterns and fault modes is essential for the model to make more robust and reliable predictions.

2.6 Conclusion

Developing a (data-driven) prognostics model is a process that requires many steps: from data acquisition required to construct the model to employing the model predictions in making better and more accurate maintenance decisions which result in less (or even zero) unscheduled downtime this reduced costs and decreased production loss, the latter being the ultimate goal of all this discussion and prognostics/preventive maintenance literature in general. The focus of this discussion is data-driven models, one of the models that proved great ability in learning complex non-linear patterns in data are neural networks. These models have different architecture depending on the structure of data. They will be presented and explained in details in the next chapter.

2.7 Conclusion

Vibration data are discrete signals sampled at a certain frequency in time. Although they hold so many valuable information about equipment performance, these information are usually not directly observable in the time domain. Digital signal processing techniques offer a way to gain more insights from raw vibration data by converting it to frequency or time-frequency domains where unusual frequency components can indicate development of certain degradation pattern. This chapter introduced several of these techniques like Fourier and Wavelet transforms. A following chapter will present the use of these techniques for extracting features that serve as an input for a neural network that can estimate the remaining useful life.

Chapter 3

Introduction to Artificial Neural Networks

Artificial neural networks are computational systems capable of finding complex functions that link an input to an output. These systems can be used for a variety of tasks such as regression and classification. They can be used in predictive maintenance and prognostics to estimate the health of equipment and predict with some uncertainty its remaining useful life. This chapter discusses neural networks, their topology, training and mathematical formulation.

3.1 Structure of artificial neural networks

Artificial neural networks are computational systems used to find the mapping between an input and an output, they consist of several layers (input layer, output layer and an arbitrary number of hidden layers between the input and the output) and each layer contains a number of neurons where each neuron in each layer is connected to all neurons in the previous and the next layer (except for the input and output layers which are connected only to the next and the previous layer respectively). Each neuron of each layer receives an input from the neurons of the previous layer (in the form of a vector), multiplies the vector by a few weights and sums the result and then applies a linear activation function. Each neuron ends up with a single numerical value called activation, which will be passed on to the neurons in the next layer.

Figure 3.1 shows a neural network with the following structure:

- Input layer with 3 inputs
- Single hidden layer with 4 neurons
- Output layer with 1 neuron

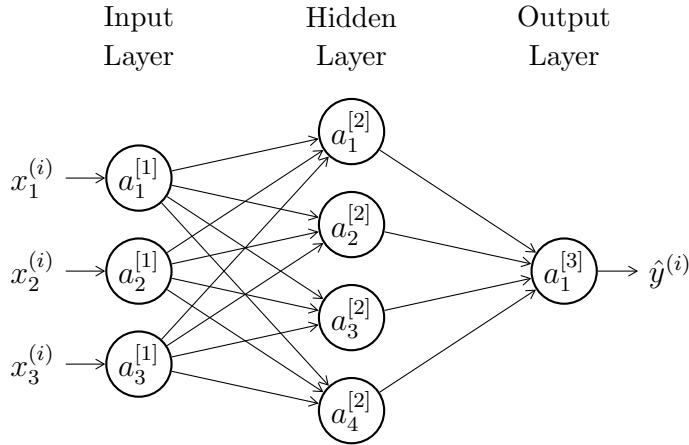


Figure 3.1: Structure of an artificial neural network

3.2 Feedforwad networks: from input to output

The architecture of figure 3.1 is called Feedforward Neural Network, it is an acyclic architecture where information flows from the first to the last layer without any internal loop, unlike other architectures such as recurrent neural networks. The first layer is the input layer, it does not perform any operation and is limited to receiving input. The input is a vector of numbers representing the different variables. The vector is multiplied by a matrix of weights that transforms it and sends it to the next layer (or to the first hidden layer). An activation function is applied to the values resulting from the multiplication of the values of the previous layer with the weight matrix, the result becomes the values of the next layer, or activations (the value of each neuron is called activation).

A general formula for moving from the $l - 1$ layer to the l layer is given by the equation 3.1:

$$a^{[l]} = g^{[l]}(W^{[l]}a^{[l-1]} + b^{[l]}) \quad (3.1)$$

Where $g^{[l]}$ is the activation function of the l layer, $W^{[l]}$ is the weights matrix that transforms the values (activations) from the $l - 1$ layer to the l layer and $a^{[l]}$ represents the activations of the l layer. b is the value of the bias (or intercept value), it is added to the multiplication between the activations and the weights matrix, it is a parameter that can be learned like the weights. The operation is repeated for each layer up to the output layer. The first layer can be considered as layer 0 and the inputs can be designated by the vector $a^{[0]}$.

3.3 Activation functions

The activation functions are applied to the result of multiplying the inputs of the previous layer with the corresponding weights to determine the value of each neuron. There are different types of these functions.

The use of the non-linear activation function is very important for neural networks,

they allow learning the complex non-linear mapping from input to output. If the network does not use non-linear activation (for example, linear activation or the identity function), then the whole network (regardless of its depth) is equivalent to a network with only one hidden layer.

There are a variety of activation functions that can be used for the hidden and output layers. Figure 3.2 shows some examples:

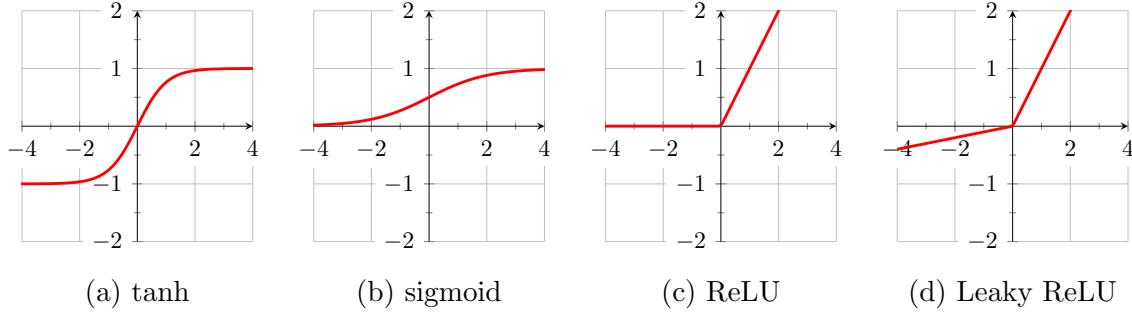


Figure 3.2: Different activation functions

Table 3.1 shows mathematical definition of several activation functions:

Activation function	Mathematical definition
Identity (no activation)	$\text{Id}(x) = x$
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$
tanh	$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$
Rectified Linear Unit (ReLU)	$\text{ReLU}(x) = \max(0, x)$
Leaky ReLU	$\text{LeakyReLU}(x) = \max(0.1x, x)$

Table 3.1: Mathematical definition of several activation functions

3.4 Network training

The process of training a neural network consists of determining the weights (coefficients) that map the neurons of each layer to the neurons of the next layer. This process can be formulated in more mathematical terms as an optimization problem: optimization of the network coefficients to find their values that minimize a cost function.

Neural network training uses training data that provide inputs and their corresponding outputs.

3.4.1 Cost function

The cost function is the function used to calculate the difference between the output of the neural network and the expected actual output, it quantifies the performance of the network. The goal of the training process is to minimize this function by using

Gradient Descent (see next section) to find the best set of weights that gives the lowest difference between the training data and the network prediction.

There are many types of cost functions, each type corresponds to different neural networks tasks (e.g. regression, binary classification, ...). The cost function usually used for regression problems is mean-squared error function (Equation 3.2) which calculates the sum of distances between the model predictions \hat{y}_i and the actual output (y_i), N is the number of data points available for training:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.2)$$

The other major task of neural networks and other types of models is classification. There are two main types of classification: binary classification, or classification of two classes and multiclass classification or classification of many classes. The first one uses binary cross-entropy loss function (Equation 3.3) and the latter uses categorical cross-entropy.

$$BCE = -\sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (3.3)$$

The choice of activation function for the last layer of the network is directly related to the used cost function. For regression tasks, linear activation function is used. For binary classification it's sigmoid and for multiclass classification it's softmax function.

3.4.2 Gradient Descent

Gradient Descent is an iterative optimization algorithm used to optimize a differentiable function. Gradient Descent works by calculating the gradients of the objective function and then taking iterative steps in the negative direction of the gradients.

3.4.3 Convex and non-convex functions

Gradient descent in neural networks is used to optimize (minimize) the cost function by finding the best set of weights and biases that give the lowest cost possible. Objective functions can be categorized into two types: convex and non-convex functions. A convex function is said to be convex if it has the property that every chord lies on or above the function. Any value of x in the interval from $x = a$ to $x = b$ can be written in the form $\lambda a + (1 - \lambda)b$ where $0 \leq \lambda \leq 1$. The corresponding point on the chord is given by $\lambda f(a) + (1 - \lambda)f(b)$, and the corresponding value of the function is $f(\lambda a + (1 - \lambda)b)$ (Figure 3.3). Convexity then implies:

$$f(\lambda a + (1 - \lambda)b) \leq \lambda f(a) + (1 - \lambda)f(b) \quad (3.4)$$

This is equivalent to the requirement that the second derivative of the function be everywhere positive [27]. This condition of convexity can be extended to spaces with arbitrarily large number of dimensions.

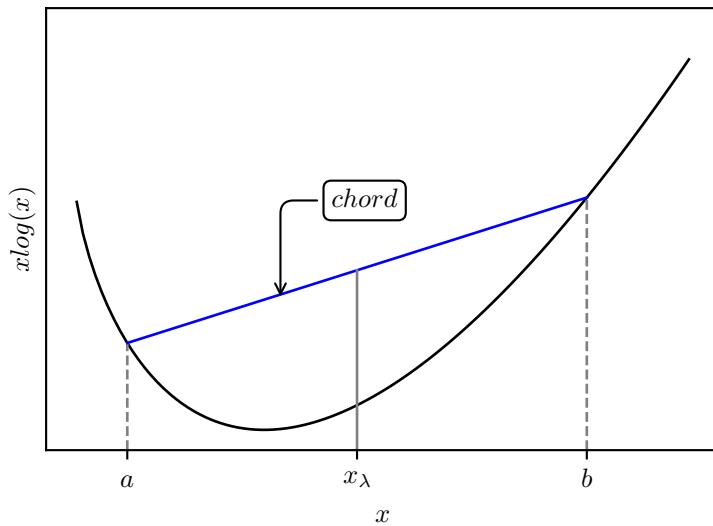


Figure 3.3: Condition of convexity

A non-convex function is a function that doesn't satisfy the convexity condition. Figure 3.4 shows an example of a convex (left) and a non-convex function (right) where the function parameters are in two-dimensional space.

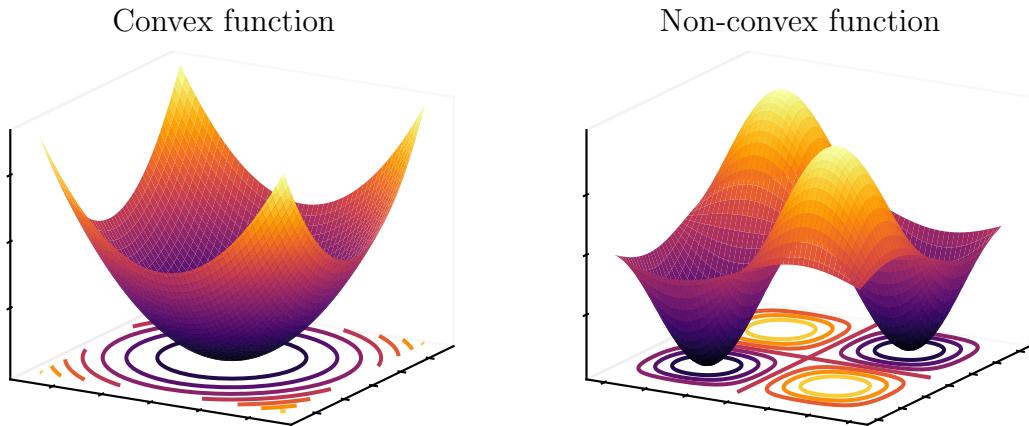


Figure 3.4: Example of convex and non-convex functions

3.4.4 Global and local minima

Global minimum refers to the lowest possible value in a set (or of a function). Finding the global minimum refers to finding the set of parameters that correspond to this minimum value. When the function is convex, finding the global maximum is possible and easy, algorithms like gradient descent always converge to the global minimum in convex optimization. Examples of convex cost functions is linear regression cost function. For more complex models such as neural networks, cost function is highly non-convex with many local minima.

Figure 3.5 shows an example of a non-convex function with a global and local minima.

The problem with non-convex optimization is that the optimization algorithm can converge to the local minimum instead of the global one. The algorithm convergence is related to the random initialization of network weights.

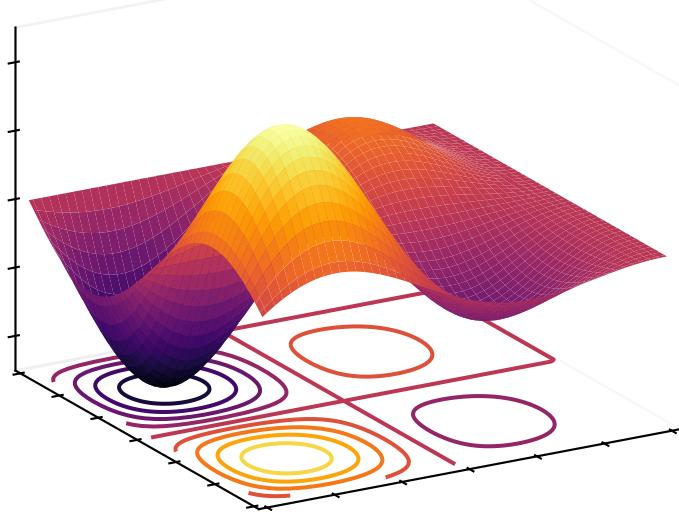


Figure 3.5: Non-convex function with global and local minima

3.4.5 Saddle points

A saddle point or minimax point is a point on the surface of the graph of a function where the slopes (derivatives) in orthogonal directions are all zero (a critical point), but which is not a local extremum of the function (Figure 3.6).

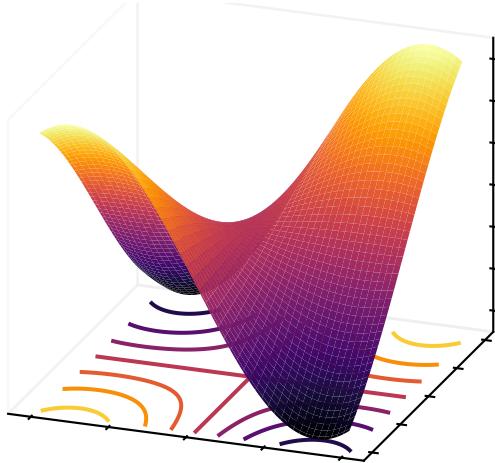


Figure 3.6: Saddle point

3.4.6 Optimization of multilayer networks

In [28] the authors showed that getting stuck in poor local minima is a major problem only for shallow networks but becomes gradually of less importance as the network

size increases. This is mainly due to the fact that in large networks, local minima lie close to the global minimum so they yield similar good performance.

The following hypotheses were also verified empirically in the mentioned paper regarding learning with large-size networks:

- When a network is large, there isn't a significant difference in performance on test data among most of local minima.
- Probability of finding a bad local minimum in small networks is higher than the probability of finding them in larger networks.
- Finding the global minimum on training set doesn't guarantee a better performance on testing data, rather it could cause overfitting and reduced performance.

3.4.7 Backpropagation

Backpropagation is the algorithm used to calculate the gradients of the cost function (the objective function) of a neural network. Since a neural network can be interpreted as functions composition, Backpropagation uses the derivation theorem of a composition of functions to find the gradients in relation to the network weights.

The use of Backpropagation for training neural networks was popularized by David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams [29], they described it as a procedure that repeatedly adjusts the weights of the network connections so as to minimize a measure of the difference between the actual output vector of the network and the desired output vector. As a result of the weight adjustments, "hidden" internal units that are not part of the input or output come to represent important characteristics of the task domain, and the regularities of the task are captured by the interactions of these units.

Figure 3.7 represents the Forward Pass and the Backward Pass, the Forward Pass calculates the network output, the Backward Pass calculates the gradients of the cost function that measures the difference between this output and the actual desired output:

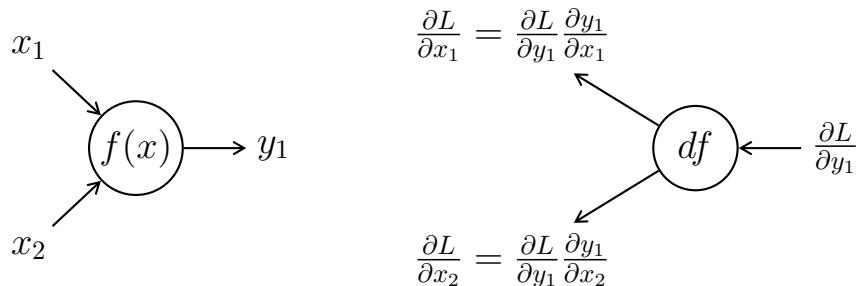


Figure 3.7: Forward (left) and Backward (right) passes

3.5 Recurrent Neural Networks

Recurrent Neural Networks (RNN) is a special architecture that is more suitable for modeling sequential data. RNNs process an input sequence one element at a time, maintaining in their hidden units a 'state vector' that implicitly contains information about the history of all the past elements of the sequence [30]. Figure 3.8 shows an example of a RNN architecture. To the left is the unfolded version with a cyclic loop in the hidden layer, to the right is shows unfolding of the network which is easier to understand. x_1, x_2, \dots, x_t represents the input vector (sequence), y_1, y_2, \dots, y_t represents the output vector (can be a sequence with length equals to the input, of different length or a single element). h_1, h_2, \dots, h_t are neurons of the hidden layer. In RNN each neuron passes a vector of its hidden state to the next neuron corresponding to the next time step.

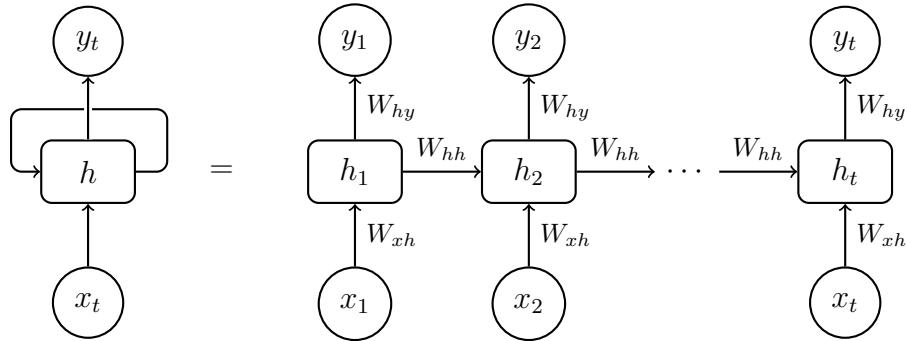


Figure 3.8: Recurrent neural networks architecture

3.5.1 Long Short-Term Memory

Simple RNNs have problems with learning long-term dependencies (i.e. making predictions based on predictions made many steps previously in the past) and vanishing gradients. Long Short-Term Memory (LSTM) was first introduced by Jürgen Schmidhuber and Sepp Hochreiter [31]. LSTM overcomes problems with basic RNN by using what is called a cell state which enables LSTM networks to learn long-term dependencies. LSTM cells also possess three types of gates which together control the flow of information inside the cell:

- **Forget gate:** Controls which information are kept or discarded at time t
- **Input gate:** Controls which information to store in the cell state at time t
- **Output gate:** Controls the final output of the cell at time t

Input, forget and output gates values are calculated using Equations 3.5, 3.6 and 3.7 respectively:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3.5)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (3.6)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (3.7)$$

$$(3.8)$$

Input, forget and output gates are calculated at time t using sets of weights and biases (W_{xi} , W_{hi} , b_i), (W_{xf} , W_{hf} , b_f) and (W_{xo} , W_{ho} , b_o) that controls how each of x_t , h_{t-1} and c_{t-1} affects the value of the gate respectively. A sigmoid function is used to convert the values to the range of 0 to 1.

At each time step t , a candidate cell state \tilde{c}_t is calculated using weights and a bias term (W_{xc} , W_{hc} , b_c) that maps the values of input x_t and previous hidden state h_{t-1} to the candidate cell state (Equation 3.9). As the name indicates, \tilde{c}_t serves as a candidate to replace current cell state c_t at time t . Cell state at time t is obtained by using the forget gate to control which information are kept from the previous cell state and input gate to control which information are kept from the candidate cell state according to Equation 3.10. Final cell output is calculated using current cell state c_t and output gate according to Equation 3.11.

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3.9)$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (3.10)$$

$$h_t = o_t \tanh(c_t) \quad (3.11)$$

Gates control the flow of information inside the LSTM cell, their values range from 0 to 1 (sigmoid function) and control which information to keep and which to discard when updating the cell state (forget and input gates) and when calculating the LSTM cell output (output gate).

Figure 3.9 illustrates the different operations happening in a single LSTM cell:

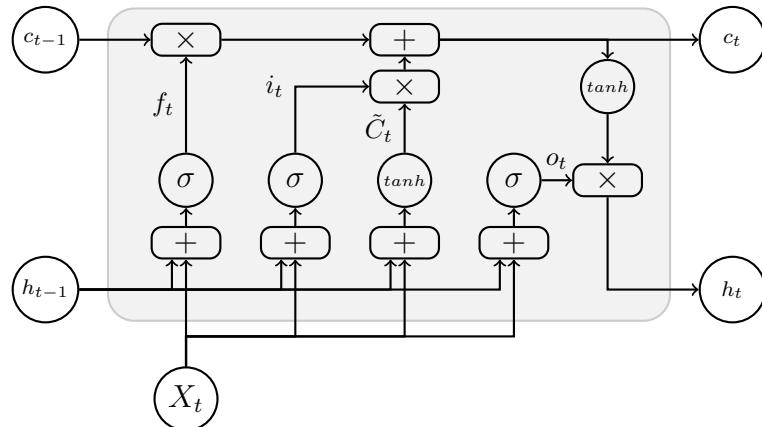


Figure 3.9: Long Short-Term Memory Cell

3.6 Convolutional neural networks

Convolutional Neural Networks (CNN) have what is called convolutional layers. A convolutional layer takes an input (usually 2D images, but it can be 1D or 3D either) and apply a mathematical operation called convolution¹. Convolutions act like filters that extract features from raw data by recursively multiplying what is called kernel with the input data. Different kernels serve different purposes and can extract a wide variety of features from raw data. After feature extraction done by the convolutional layer, they serve as an input for a feedforward neural network (as described in section 3.2) for classification or regression.

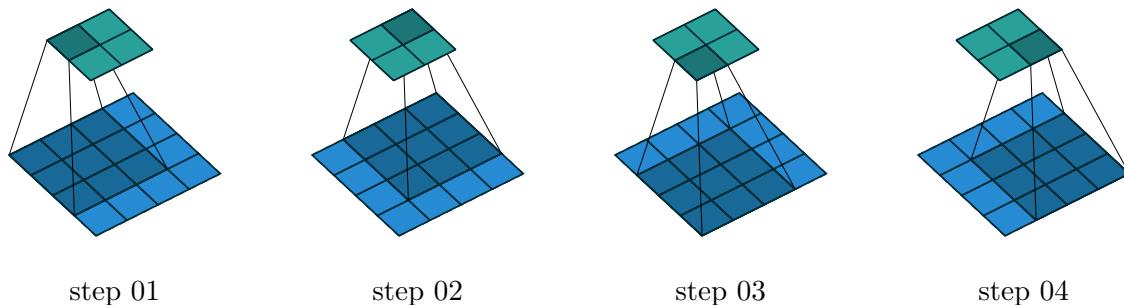


Figure 3.10: Applying convolutions (by multiplying the input by a kernel) iteratively to different regions (dark blue) of the 2D input (whole blue square). Each iteration gives a numerical value (dark green). The result of all iterations is the green 2D output [3].

3.7 Conclusion

Neural networks are a powerful tool to find the complex relationship between an input and an output (e.g. Condition Monitoring data and machine degradation). They consist of different layers of interconnected neurons. the connection between each 2 layers is defined by a set of weights, which are multiplied by the values of the previous layer to find the values of the next layer, the process is repeated until the output layer is reached. the predicted output is compared to the actual output obtained from the training data, the weights are adjusted to minimize (using Backpropagation and Gradient Descent) the difference between the predictions and the reality.

¹Mathematically, the operation happening in a CNN is called cross-correlation which is a bit different than the mathematical definition of a convolution. In Machine Learning literature those operations are called convolutions, this is the terminology that will be used here.

Chapter 4

Equipment Health Assessment using Artificial Neural Networks

This chapter demonstrates the use of an artificial neural network to estimate the health state of a set of turbofan engines from NASA C-MAPSS dataset and predict the remaining useful life. Also different metrics to assess the performance of the network will be introduced.

4.1 Introduction to NASA C-MAPSS dataset

C-MAPSS is a tool for simulating a realistic large commercial turbofan engine. The software is coded in the MATLAB® and Simulink® environment.

C-MAPSS software consists of many editable input parameters that controls the simulation, these inputs are specified by the user and control many aspects of the simulation such as operational profile, closed-loop controllers, environmental conditions, etc. [4].

Figure 4.1 is a simplified diagram of the simulated engine showing its main elements, like low pressure compressor section (LPC), high pressure compressor section (HPC), fan and combustor. The dataset released by NASA Ames Research Center contains resulting data from simulating many turbofan engines,

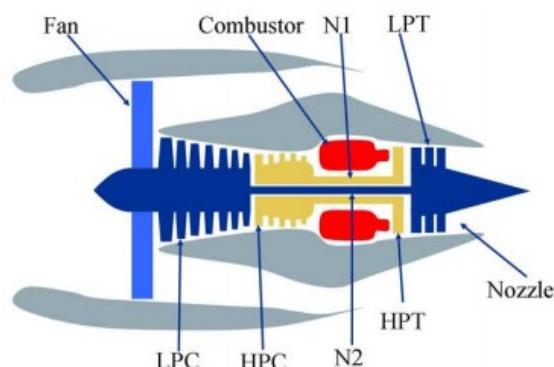


Figure 4.1: Simplified diagram of engine simulated in C-MAPSS [4]

from beginning of operation until failure. The dataset was originally released for Prognostics and Health Management 2008 Data Competition, Table 4.1 shows different variables, the output of the simulation and their units, that were provided for the participants in the competition:

Symbol	Description	Units
T2	Total temperature at fan inlet	R
T24	Total temperature at LPC outlet	R
T30	Total temperature at HPC outlet	R
T50	Total temperature at LPT	R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P_{50}/P_2)	–
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps_{30}	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	–
farB	Burner fuel-air ratio	–
htBleed	Bleed Enthalpy	–
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/

Table 4.1: C-MAPSS outputs to measure system response.

C-MAPSS data contains 4 datasets: FD001, FD002, FD003 and FD004. Each dataset has different working conditions and fault modes. Table 4.2 contains statistics of the different datasets:

	units number	max length	average length	min length
FD001	100	362	206.31	128
FD002	260	378	206.77	128
FD003	100	525	247.2	145
FD004	249	543	245.95	128

Table 4.2: Units number and cycles length statistics in C-MAPSS data

4.2 Visualization of equipment degradation

Visualizing the simulation output can give a sense of how these variables change during the life of the engine, Figure 4.2 shows four different sensors from one of the engines (values are normalized):

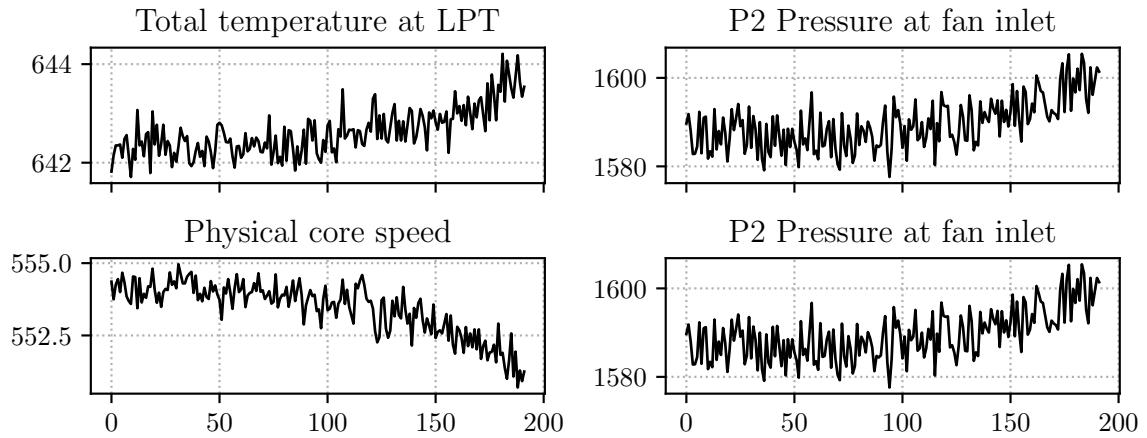


Figure 4.2: Development of 6 sensors outputs from one of the engines (normalized)

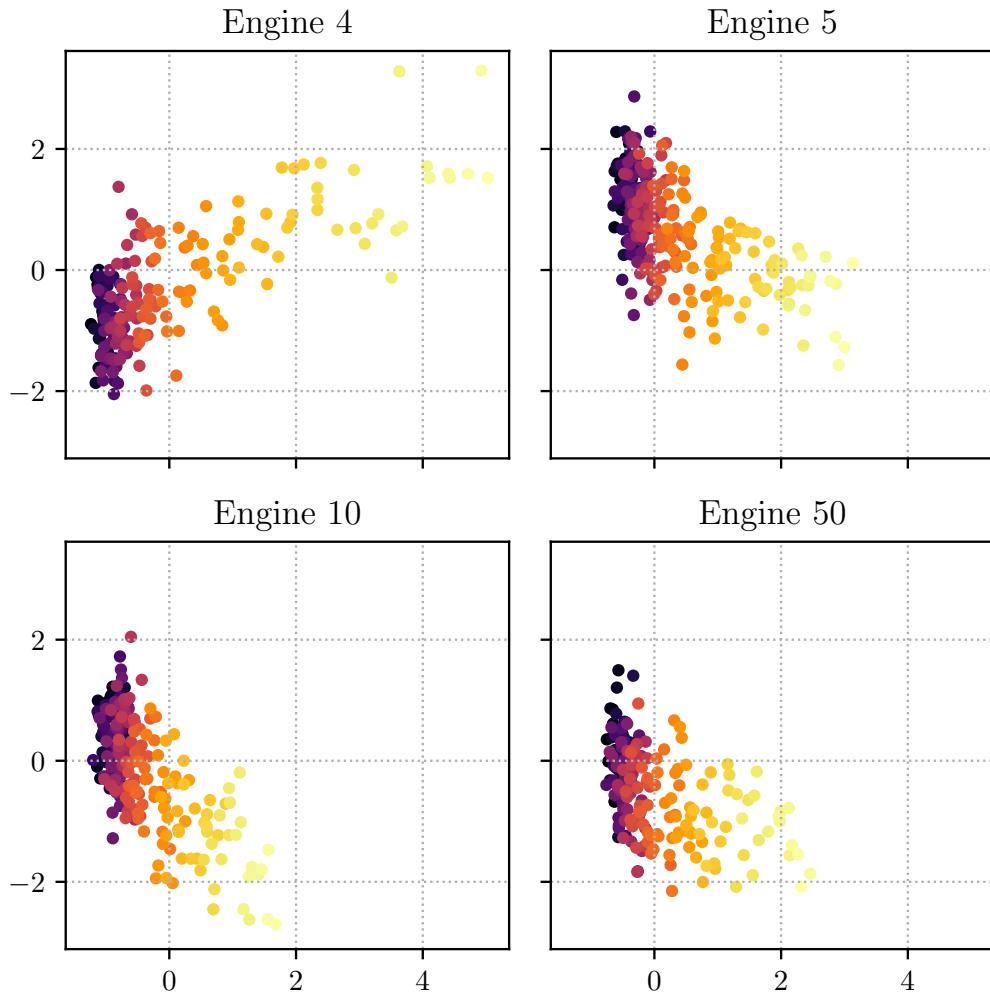


Figure 4.3: Equipment health degradation (lighter colors indicate advancing of health degradation) of four turbofan engines from C-MAPSS dataset

It is apparent that sensors output follow a specific pattern (increasing or decreasing)

from beginning of operation until breakdown, this is very useful and can increase the robustness of the predictive model.

Alternatively, all sensors values can be combined and visualized altogether using Principal Component Analysis (section 2.2.7) to reveal the general trend in the data, if condition monitoring data is directly indicative of the equipment health state, visualization of principal components can show apparent visual degradation patterns.

Sensors values from 4 different engines are combined using PCA and the two first principal components are represented on Figure 4.3.

There is absolutely an apparent pattern for health state degradation across the different engines from left (where darker colors indicate normal working state) to the right (where the lighter colors indicate fault development).

4.3 Engines health classification

Before proceeding with a complicated task such as RUL estimation, a simpler task like engines health classification can show the complexity of the problem. This section describes using neural networks to classify engines states as healthy or faulty. The first and last 25 cycles from each unit are considered to be healthy and faulty respectively.

A neural network that uses all 24 inputs (all operational settings and sensors) with two hidden layers is used to carry out this classification. Table 4.3 summarizes the network architecture:

Layer (type)	Output shape	Param #
Dense1 (Dense)	(None, 8)	200
Dense2 (Dense)	(None, 4)	36
Dense3 (Dense)	(None, 1)	5
Total params: 241		
Trainable params: 241		
Non-trainable params: 0		

Table 4.3: C-MAPSS classifier architecture

The model is trained for 200 epochs with batch size of 32 samples, the classifier achieves **93.46% accuracy** on the test set. Figure 4.4 shows the training process of the network. In the upper plot, the y-axis corresponds to the train and validation losses (binary crossentropy). The y-axis in the lower plot corresponds to the network train and validation accuracies, the x-axis shared between the two plots indicates the training epochs. Although the validation accuracy varies a lot, the training accuracy keeps improving with each epoch.

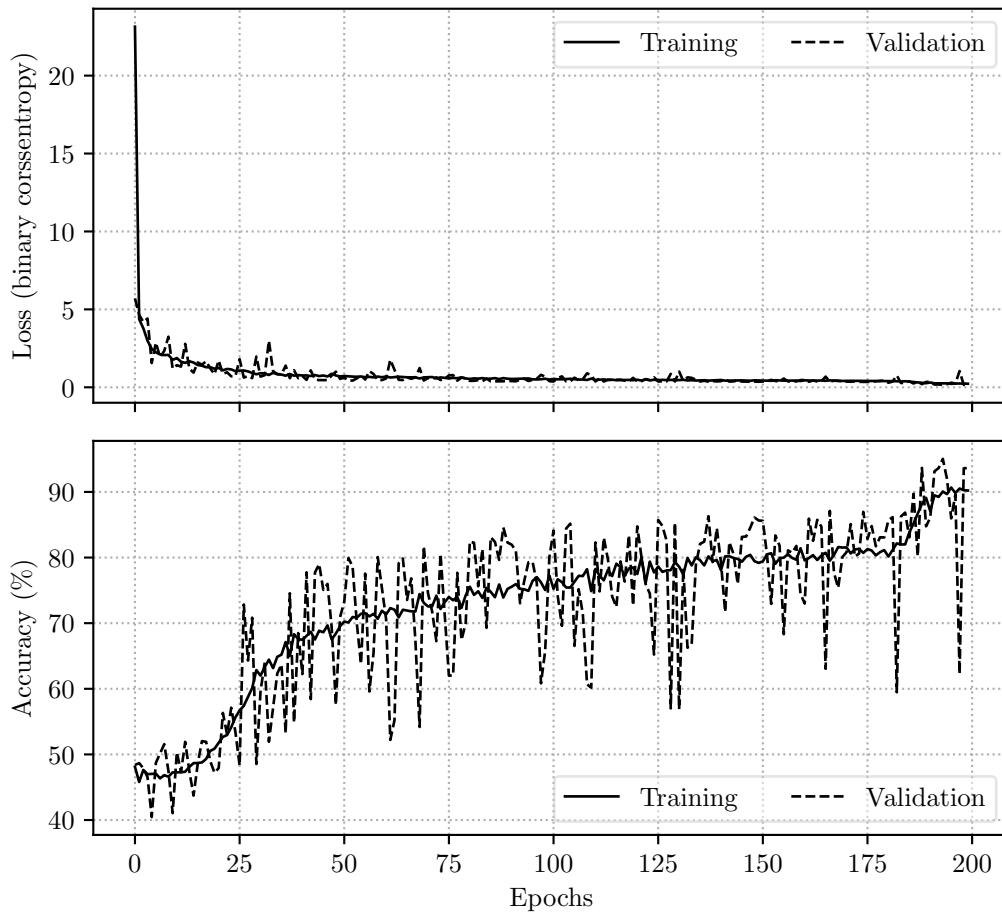


Figure 4.4: Training process of C-MAPSS classifier

Figure 4.5 shows Receiver Operating Characteristics (ROC) Curve of the classifier with a high Area Under Curve (AUC). Other classification metrics are shown in Table 4.4.

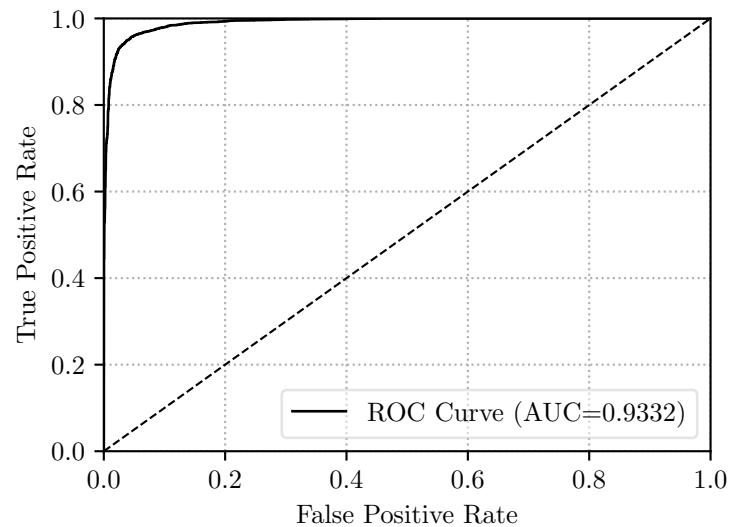


Figure 4.5: C-MAPSS classifier ROC curve on test set

Metric	Accuracy	Precision	Recall	F-1	ROC AUC
Score	93.43%	0.90	0.98	0.94	0.9332

Table 4.4: C-MAPSS classifier metrics on test set

4.4 Remaining Useful Life prediction

4.4.1 Remaining Useful Life modeling

In order to train a neural network to estimate the Remaining Useful Life (RUL) of a new unseen data from C-MAPSS dataset, an appropriate RUL that corresponds to the training data must be constructed. RUL was defined in Section 1.2, from this definition several approaches to construct an appropriate RUL can be developed for units in the train data (where the total number of cycles before failure is known). The simplest approach is to use an always-decreasing RUL, this implies that the equipment state is always decreasing and moving towards failure. The problem with this approach is that degradation process isn't linear and in real life, machines don't start to degrade as soon as they are start working. A second approach is to use a piecewise function where the equipment state is constant at first then at a specific point it starts to degrade linearly. This approach is much better than the first one but it is not very representative of the real world behavior of degradation process. In the context of this thesis, RUL is approached as a nonlinear polynomial function where the degradation is slow at first then accelerates towards the end of life. Figure 4.6 shows the different choices for modeling RUL:

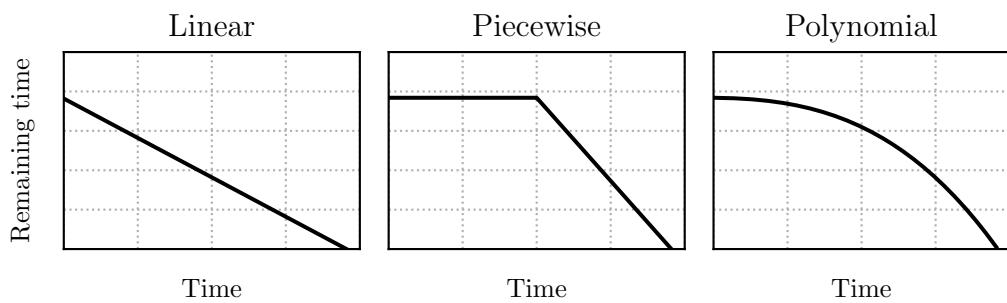


Figure 4.6: Different RUL modeling choices

It must be noted that not any of these approaches can be claimed to be representative of the real world degradation process, but choosing the most intuitive representation of RUL can help the learning algorithm learn the implicit degradation trends in the data.

4.4.2 RUL prediction using feed-forward network

The last section dealt with classifying units health state as healthy or faulty. This section instead presents a neural network for estimating the RUL. In this section, a

neural network architecture with 3 hidden layers is used. Since this is a regression problem, mean squared error is used as the loss function, mean absolute error is used as a metric for the model. Table 4.5 presents the architecture details:

Layer (type)	Output shape	Param #
Dense1 (Dense)	(None, 32)	800
Dense2 (Dense)	(None, 16)	528
Dense3 (Dense)	(None, 8)	136
Dense4 (Dense)	(None, 1)	9
Total params:	1,473	
Trainable params:	1,473	
Non-trainable params:	0	

Table 4.5: Fully-connected network architecture for RUL prediction

The network was trained for 300 epochs, batch size of 128 samples. Figure 4.7 shows the training process of the network. The figures show the development of training and validation losses (mean squared error) and metrics (mean absolute error) respectively as a function of training epochs.

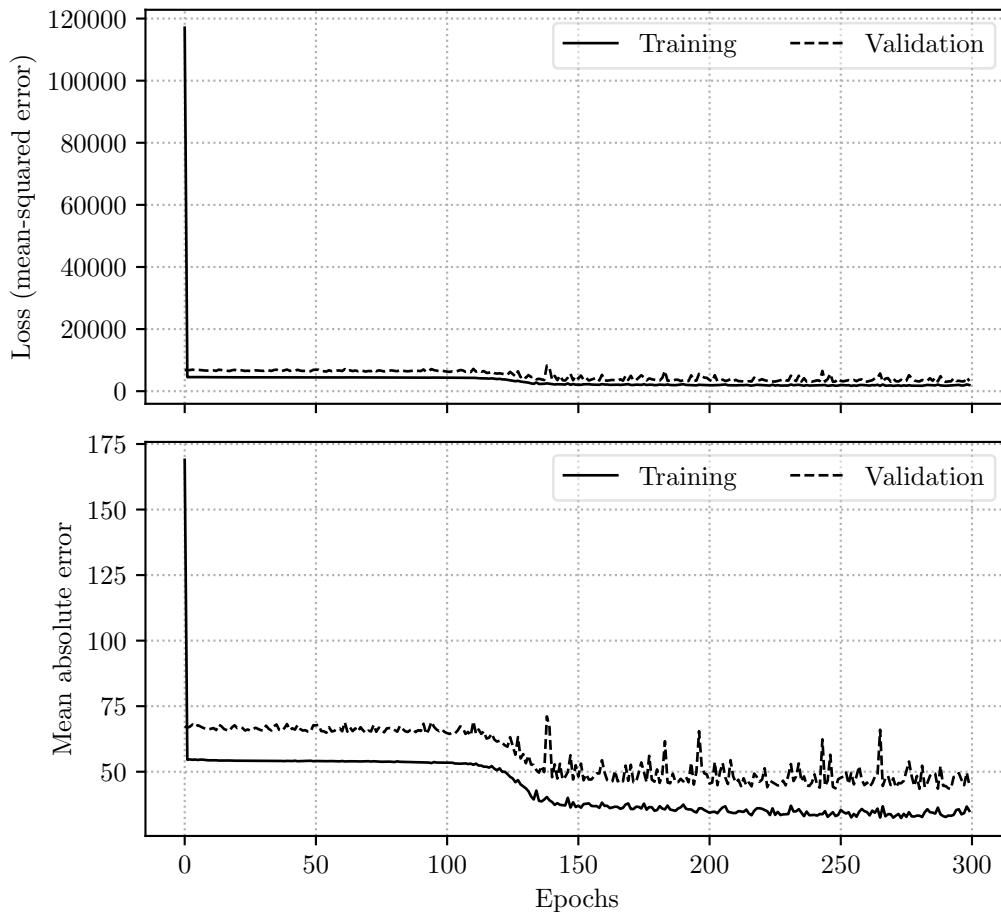


Figure 4.7: Training process of RUL predictor

After training, the model is evaluated on two units from the test set. Figure 4.8

shows the actual RUL (dashed line), model prediction at each cycle and a 3rd degree polynomial fit of model's predictions.

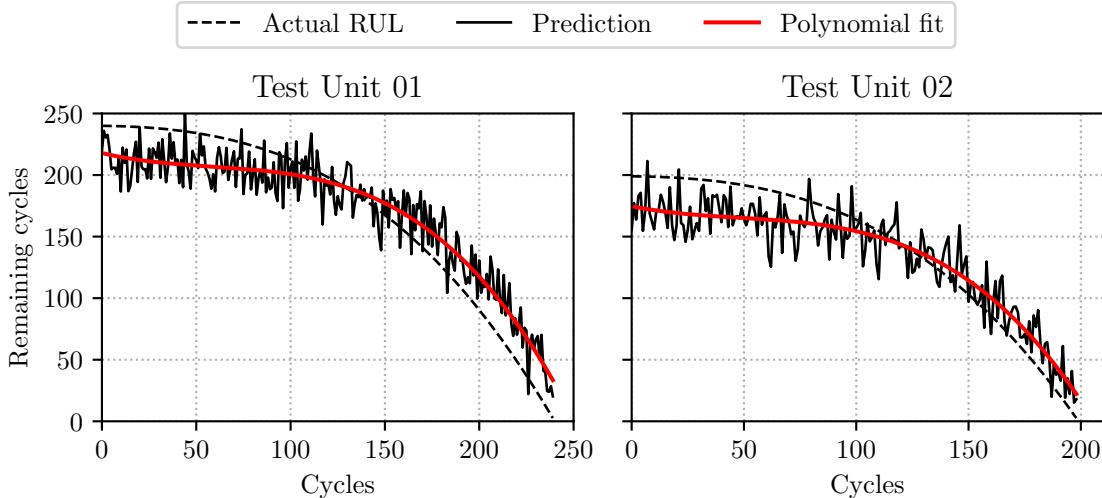


Figure 4.8: Prediction results on two test units

Although the model's prediction are close the the actual RUL, but they are noisy and there is so much fluctuation. The noise can be reduced by smoothing the output for example using moving average or fitting the points to a polynomial. But the reason of the noise in the first place is that fully-connected architecture doesn't take into consideration the previous predictions (i.e. acyclic architecture). Since RUL at each instant of time depends on the RUL from the previous one, taking previous steps into consideration while making new predictions can effectively reduce the noise and also improve prediction accuracy. To achieve that, a cyclic neural architecture like recurrent neural networks should be used.

4.4.3 Improving RUL prediction using LSTM networks

Fully-connected neural networks are powerful tool for modeling a wide range of problems, but using the fully-connected architecture for time series prediction, such as RUL prediction, can yield very noisy output because the network is acyclic and each instant in time is evaluated separately and doesn't take into account the previous predictions. Long Short-Term Memory (Section 3.5.1) are a powerful tool for modeling such problems and to make more robust and less noisy predictions. In this section, fully-connected neural network architecture from previous section is replaced with an LSTM network to predict RUL in C-MAPSS dataset.

The architecture used here is described in Table 4.6:

It is apparent that the LSTM architecture has much more parameters (216,730 parameters) than fully-connected architecture (1,473). This is because of the more complicated design of LSTM cells and their possession of different gates. This results in much longer training time.

LSTM layers take 3 dimensional tensor as an input of shape (*samples, sequence length, features*). Sequence length was set to 100 in this architecture, that's why every training

Layer (type)	Output shape	Param #
LSTM1 (LSTM)	(None, 100, 100)	50000
LSTM2 (LSTM)	(None, 100, 100)	80400
LSTM3 (LSTM)	(None, 75)	52800
Dense1 (Dense)	(None, 120)	9120
Dense2 (Dense)	(None, 110)	13310
Dense3 (Dense)	(None, 100)	11100

Total params: 216,730
 Trainable params: 216,730
 Non-trainable params: 0

Table 4.6: LSTM network architecture for RUL prediction

sample must have a shape of *(sequence length, features)*. All units have the same number of features but different lengths, since every sample must have a sequence length of 100, samples with cycles less than 100 are padded with -10 where LSTM layers are set to ignore any time steps with this value.

The network was trained for 50 epochs with batch size of 16 and validation split of 0.2 using Adam optimizer. Training process is visualized in Figure 4.9:

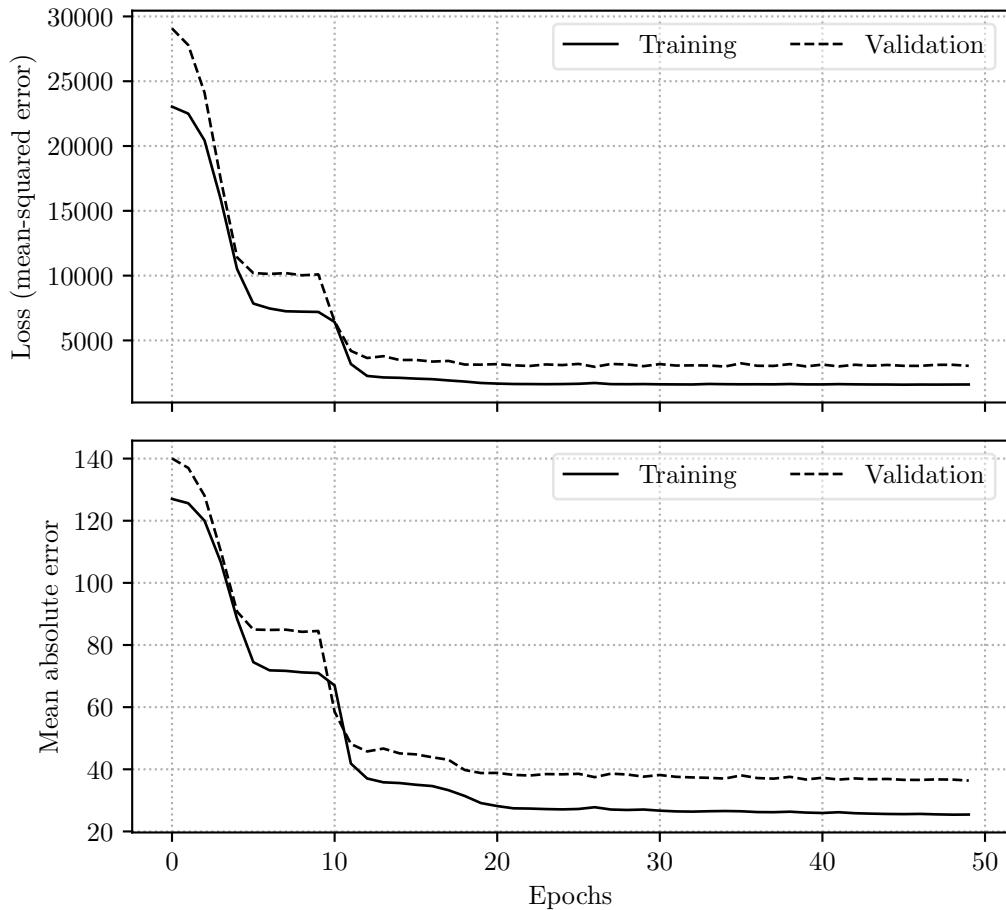


Figure 4.9: LSTM training process

Table 4.7 shows the loss and metric (i.e. mean absolute error) on the different train, validation and test sets:

	Loss	Mean Absolute Error
Train set	1600.60	25.43
Validation set	3039.68	36.37
Test set	1379.67	23.27

Table 4.7: LSTM training results

Four different units were reserved as testing units, after training the network is used to predict RUL on test units. Figure 4.10 shows prediction results and actual RUL of two different units.

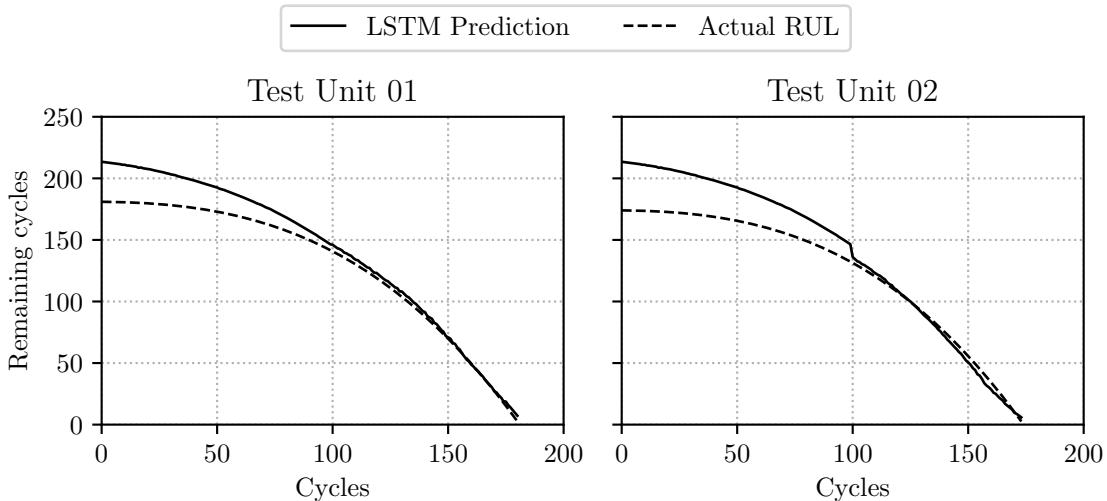


Figure 4.10: LSTM prediction results on two test units

It's apparent that LSTM network predictions are much better and almost free of noise compared to those made by the fully-connected network shown in Figure 4.8. Predicted RUL is almost identical to actual RUL towards the end of life of the unit.

4.5 Application to oilfield equipment

This chapter presented a predictive maintenance approach for RUL estimation based on condition monitoring data provided by different sensors that can measure different physical variables that are related to the equipment's health state. The data used in the previous sections consisted of C-MAPSS turbofan engines dataset, but the same approach can be extended to other applications like oil rigs.

Modern oil rigs contain a large number of sensors distributed across almost every critical equipment. These sensors measure different physical variables such as temperature, flow, pressure.... But the majority of the data provided by these sensors isn't properly exploited into an appropriate framework of prognostics and predictive maintenance, there are enormous amounts of historic data in oil rigs that if exploited can greatly

enhance maintenance programs in such critical applications where reducing downtime is very important and equipment unavailability can have cause significant production losses.

Figure 4.11 shows different spots where sensors are installed on most modern oil rigs. Table 4.8 lists types of different sensors that can be used for prognostics, the units to which they are attached and which physical variables they can measure. A neural network can be trained on the rig historic data and be used to monitor different equipment, their interaction, detect anomalies and report them in time in order to carry on the appropriate maintenance actions.

4.6 Conclusion

Fully-connected neural networks are a powerful tool for quantifying health state of complex systems using condition monitoring data. C-MAPSS dataset is an example of a system with many interacting parts that provide a variety of condition-monitoring data (e.g. temperature, pressure, rpm, . . .) where the degradation isn't directly related to one component but is indicated by the sum of the trends in all the monitoring data. Neural networks are able to capture such trends and predict the RUL of the system before failure. Fully-connected networks were first used for this task, but a cyclic architecture like LSTM can produce better results with improved accuracy and much less noise.

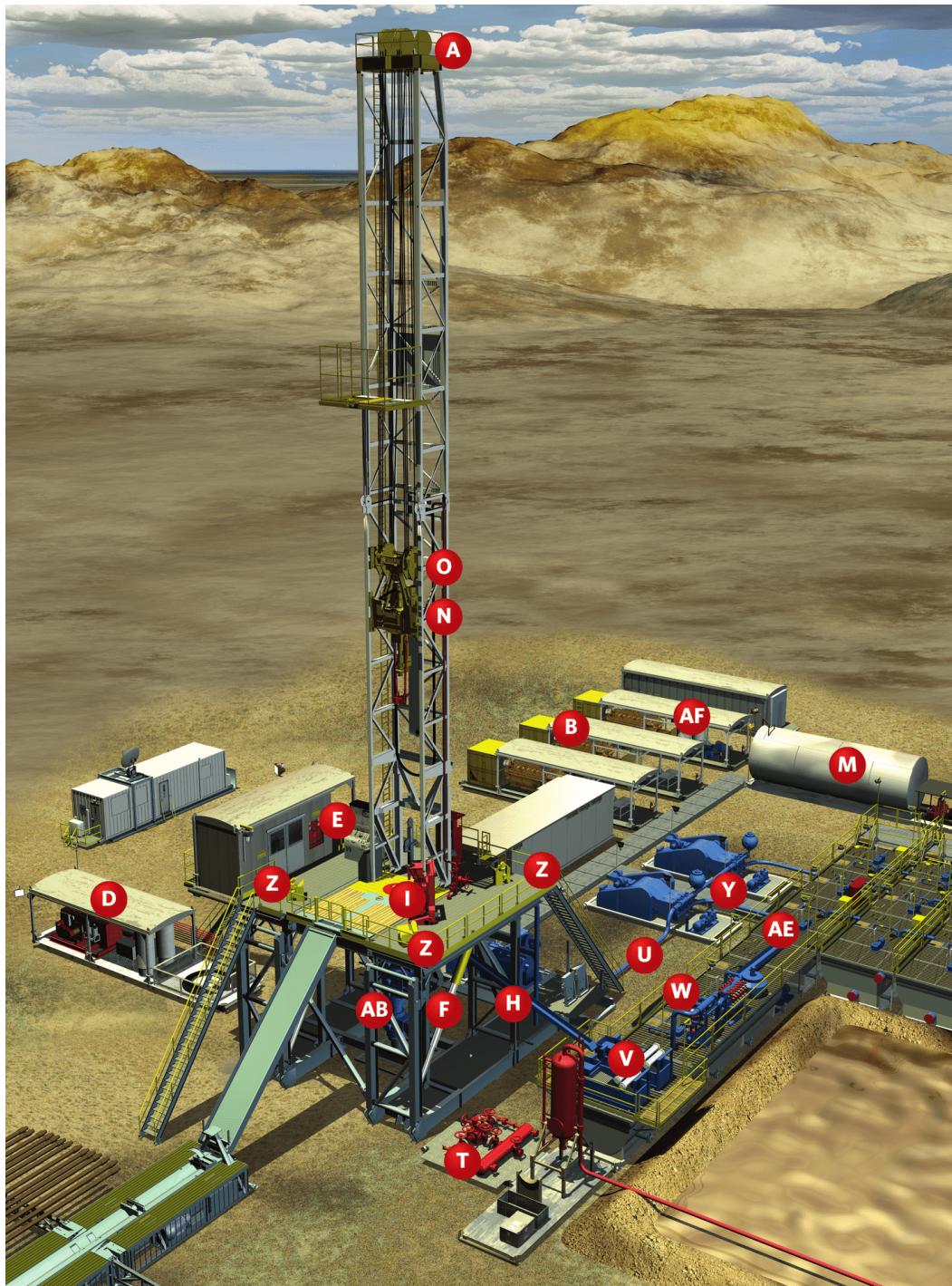


Figure 4.11: List of sensors installed on oil rigs for monitoring

Sensor	Equipment	Sensor Type	Measured quantity
A	Crown Block	Load cell	Weight on drill line via cable tension
B	Power Generation Unit	Pressure	Oil, water, and hydraulic fluid pressure
D	Accumulator Unit	Pressure	Inlet/outlet pressure with high accuracy
F	Rig Hydraulic Lift	Pressure	Hydraulic pressure, weight, force/strain, or movement, monitor raising or lowering deck for directional drilling
H	Drawworks	Load cells	Torque, load/weight/position while guiding pipe into position
M	Water/Storage Tank	Pressure	Tank liquid levels
N	Top Drive	Torque/Pressure	Monitor torque/twisting movement to ensure right amount of force is applied. Weight on drill bit. Hydraulic pressure and feed information into control system.
O	Traveling Block	Load cells	Weight on the drill line via cable tension
R	Deadline anchor	Load cells	Tension on deadline/drilling line cable
U	Mud Return Line	Pressure	Drilling mud pressure to monitor and control mud flow
Y	Mud Pump	Pressure	Pressure and flow of mud media
AB	Blowout Preventor	Pressure	Monitor RAM position via hydraulic volumetric or pressure behind the piston
AD	Drill Bit	Pressure	Pressure or differential pressure at high temperature and pressure ranges
AE	Fluid manifold	Pressure	Drilling fluid pressure
AF	Mud Tank/Reservoir	Pressure	Tank liquid levels

Table 4.8: Honeywell oil rig sensors list

Chapter 5

Bearings Faults Diagnostics and Prognostics

Vibration condition monitoring are vital for many industrial systems, vibration data contains very useful information about health state of the equipment and the fault type. Nevertheless, gaining insights from vibration signals in real-world applications turns out to be a complex—and in many times, unfruitful—process. This is mainly due to the complexity of the problem. This chapter introduces several approaches where neural networks are used for diagnostics and prognostics of bearings elements

5.1 Case Western Reserve University Bearings Data

The dataset used in this section is bearings vibration dataset provided by Case Western Reserve University (CWRU). The bearings used in the test are SKF ball bearings. Figure 5.1 shows the different components of a standard ball bearing. The test was conducted where the bearings support the shaft of a 2hp motor at different loading conditions.

The test bearings have single point faults which were introduced using electro-discharge matching with fault diameters of 0.18mm, 0.36mm, 0.53mm, 0.71mm and 1.02mm. These faults were introduced in the bearing's ball, inner and outer raceways. SKF bearings were used for the 0.18mm, 0.36mm and 0.53mm faults, and NTN equivalent were used for the 0.71mm and 1.02mm faults. Table 5.1 con-

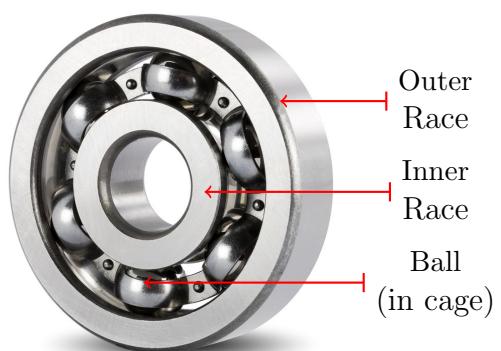


Figure 5.1: Bearing's components

tains the used SKF bearings' model dimensions and the corresponding frequencies (as multiples of RPM) associated with different faults types.

Dimension	Size (mm)	Defect	Frequency (\times RPM Hz)
Inner diameter	25.00	Inner Ring	5.4152
Outside diameter	52.00	Outer Ring	3.5848
Thickness	15.00	Cage Train	0.3983
Pitch diameter	08.03	Rolling Element	4.7135

Table 5.1: CWRU bearings dimensions and faults frequencies

5.2 Generating data from raw vibration signals

Raw vibration data can't be used directly as an input to a neural network. This chapter uses the approach proposed in [32] to convert raw vibration data into images. Figure 5.2 shows data generation principal where 1-dimensional vibration signals are converted into 2-dimensional arrays (images) by reshaping chunks of length 4096 into matrices of 64×64 .

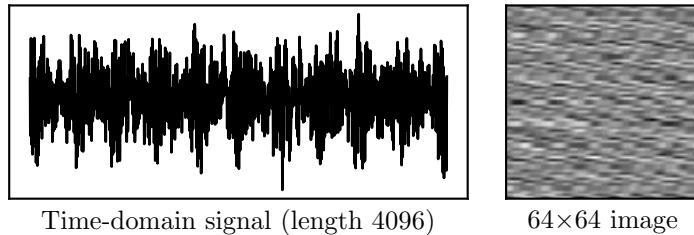


Figure 5.2: Data generation by converting the signal into 64×64 images

As mentioned before, several tests have been conducted with different types of bearings faults (i.e. ball, inner and outer raceways faults) with different fault diameters. Signals from the different tests are transformed into images to serve as an input for a convolutional neural network which will be trained to classify the different signals into the corresponding fault types and their diameters. Signals are distributed across 10 classes: three different fault types and for each fault type there are three different fault diameters, plus a normal baseline signal that belongs to healthy bearing. Figure 5.3 shows some samples of the transformed signals of the different nine fault types/diameters:

Signals of the different fault types have varying lengths which results in a different number of synthesized images per fault type. Number of images per class is given by equation 5.1:

$$N = \text{floor} \left(\frac{\text{signal length}}{64 \times 64} \right) \quad (5.1)$$

Number of images corresponding to each class is shown in Table 5.2:

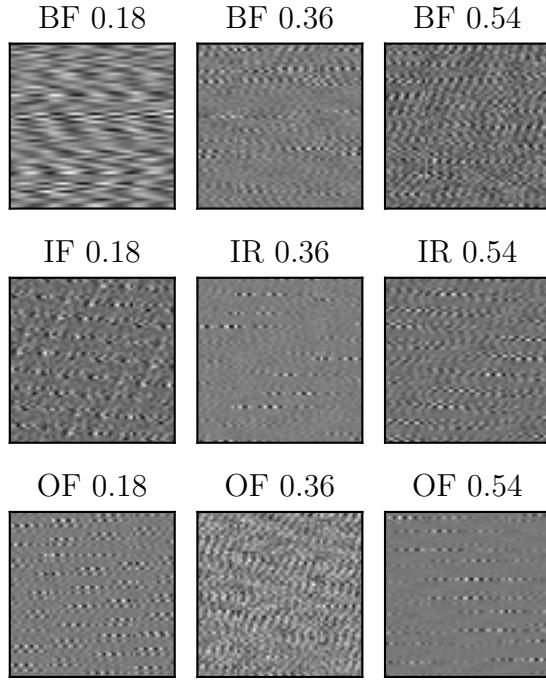


Figure 5.3: Converted signals of different faults types

5.3 Case study: Bearings faults diagnostics using neural networks

After generating data by converting raw vibration signals into images, these images serve as an input for a convolutional neural network.

5.3.1 Network architecture

Convolutional Neural Networks (CNN) describes a type of neural networks that is suitable for image processing. This section presents the use of CNN to classify bearings vibration signals that have been transformed into images into the corresponding fault types. CNN were described in details in section 3.6.

To perform this classification task, a CNN architecture is used, the network consists of three convolutional layers with max-pooling layers in between, followed by three fully-connected layers. All the details of this architecture from are mentioned in Table 5.3.

5.3.2 Training process

The dataset contains in total 1548 samples, 20% is used as a test set while the remaining 80% is the training set. From the train set, 15% is used as a validation split during the training process. The network was trained for 30 epochs and a batch size of 32. Figure 5.4 shows the evolution of the accuracy (%) and loss (categorical cross-entropy)

Class	Samples count
Normal bearing	295
Roller element 0.18mm	146
Roller element 0.36mm	116
Roller element 0.54mm	116
Inner race 0.18mm	295
Inner race 0.36mm	116
Inner race 0.54mm	116
Outer race 0.18mm	116
Outer race 0.36mm	116
Outer race 0.54mm	116

Table 5.2

Layer (type)	Output shape	Param #
Conv1 (Conv2D)	(None, 1, 64 ,32)	18464
MaxPool1 (MaxPool2D)	(None, 1, 32, 32)	0
Conv2 (Conv2D)	(None, 1, 32, 64)	18496
MaxPool2 (MaxPooling2D)	(None, 1, 16, 64)	0
Conv3 (Conv2D)	(None, 1, 16, 128)	73856
MaxPool3 (MaxPooling2D)	(None, 1, 8, 128)	0
Flatten1 (Flatten)	(None, 1024)	0
Dense1 (Dense)	(None, 128)	131200
Dense2 (Dense)	(None, 64)	8256
Dense3 (Dense)	(None, 10)	650
Total params:	250,922	
Trainable params:	250,922	
Non-trainable params:	0	

Table 5.3: CNN architecture

as a function of training epochs. The network converges around 10th epoch with minor increase in accuracy and decrease in loss afterwards.

5.3.3 Network results discussion

The model achieves perfect 100% accuracy on training set and a close accuracy on validation set. After training the model is evaluated on the test set and achieves a near perfect accuracy of 98.72% accuracy. Train, validation and test loss and accuracy are summarized in Table 5.4.

	Loss	Accuracy
Train set	0.0003	100.00%
Validation set	0.0269	99.46%
Test set	0.0586	98.71%

Table 5.4: CNN training results

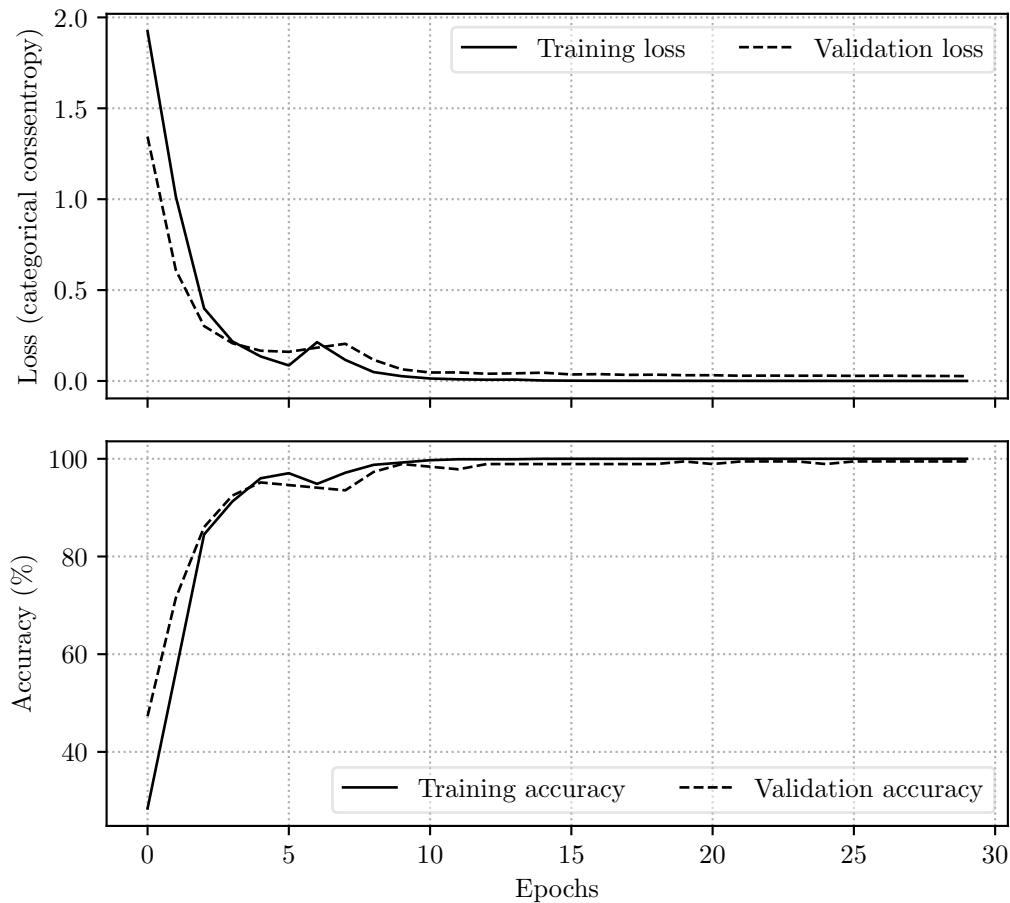


Figure 5.4: Classifier training

To further understand the model's results, confusion matrix is constructed. The confusion matrix (Figure 5.5) shows the results of model predictions on the test set where y-axis shows the actual (real) labels in the test set and x-axis the model predictions. Diagonal of confusion matrix is where actual labels intersect with model's predicted labels for each sample and it is apparent that the model achieves near-perfect classification.

5.4 Case study: Bearings faults prognostics using neural networks

Section 5.3 presented an approach for diagnosing bearings faults using vibration data and neural networks. Diagnosis is important aspect of maintenance in general, but it's not enough alone, in a predictive maintenance program and in the context of PHM, it is also important to monitor the equipment, estimate its health state and project it into the future to predict the RUL and that is what is covered in this section. In this section, FEMTO bearings dataset is used.

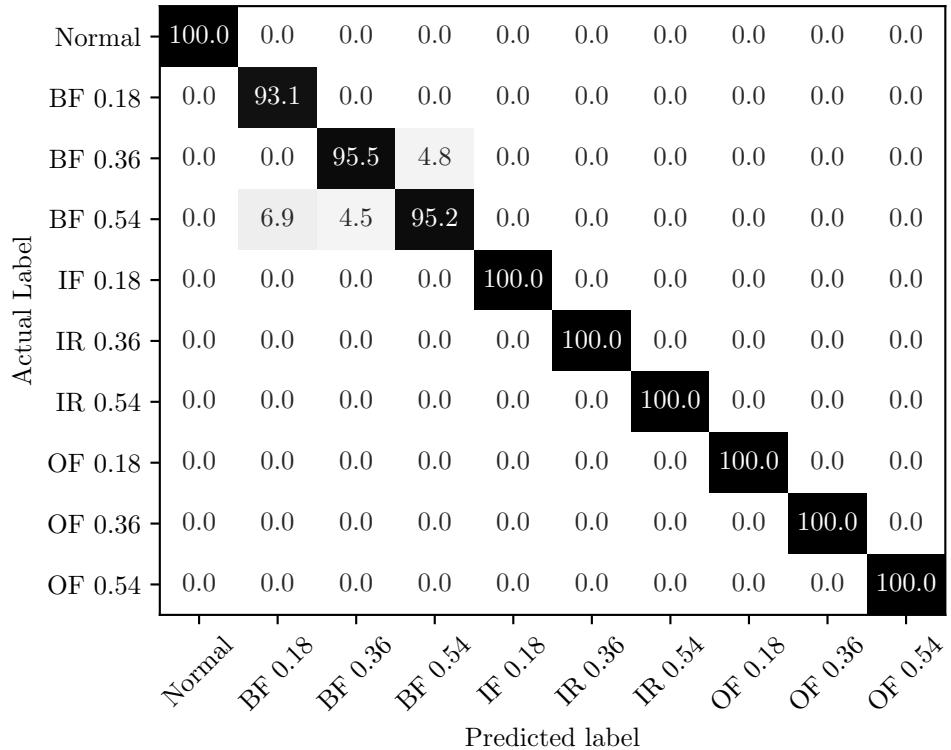


Figure 5.5: Confusion matrix of bearings faults classification using CNN

5.4.1 Introduction to FEMTO database

FEMTO Bearings dataset was provided by FEMTO-ST Institute (Besançon - France) as a part of PHM 2012 Data Challenge. The dataset consists of vibration and temperature measurements from bearings accelerated aging (run to failure) experiments which were conducted using PRONOSTIA platform.

PRONOSTIA (Figure 5.6) is an experimental platform used for accelerated bearings life tests. PRONOSTIA makes it possible to conduct these tests in few hours. Data provided by the platform corresponds to naturally degraded bearings rather than using bearings with artificially-induced faults (e.g. CWRU bearings data).

PRONOSTIA consists of three main parts:

- **Rotating part:** Asynchronous motor with a power of 250W that transmits rotation motion through a gearbox which allows a rotation speed up to 2830 rpm. The motor's speed and direction of rotation are controlled by the user through a human machine interface (HMI).
- **Load part:** Which is used to apply a radial force to induce faults in bearings. By applying a load that exceeds the bearing's maximum dynamic load (4000 N), this part allows to drastically reduce the bearing's life and conduct the accelerated aging test. The load is generated using a force actuator that consists of a pneumatic jack powered by a pressure delivered through a digital electro-pneumatic regulator.
- **Measurement part:** Operation conditions consist of three variables: a) the

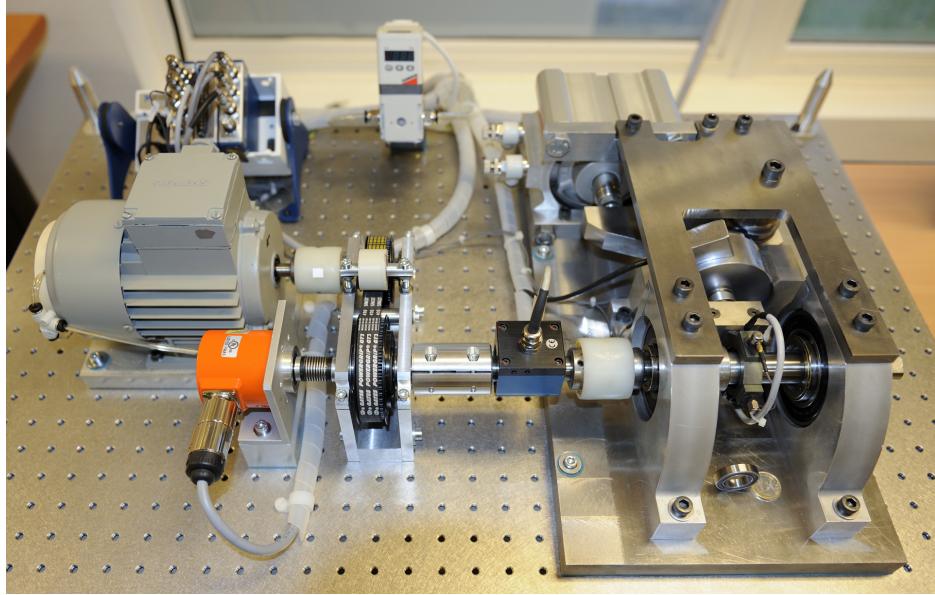


Figure 5.6: PRONOSTIA platform [5]

applied radial force, b) shaft speed and c) torque inflicted on the bearings, all three which are sampled at a frequency of 100Hz. Bearings health is measured by two variables: a) vibrations (both horizontal and vertical) which are measured by 2 accelerometers placed on the bearing's outer race and b) temperature measured by an RTD (Resistance Temperature Detector) probe placed close to the outer race. Vibration is sampled at 26.5kHz and temperature at 10Hz. Vibration measurements consist of snapshots of 2560 samples taken every 10s while temperature measurements are 600 samples taken each minute.

The bearings in the datasets are grouped under three operation conditions:

- First operating conditions: rotation speed of 1800 rpm and load of 4000N.
- Second operating conditions: rotation speed of 1650 rpm and load of 4200N.
- Third operating conditions: rotation speed of 1500 rpm and load of 5000N.

Table 5.5 shows different operating conditions and their corresponding bearings in both learning and testing datasets:

5.4.2 Generating data from scaleograms

It is important to detect bearing degradation before the actual failure happens. In this section, a convolutional neural network architecture will be used to classify healthy and faulty bearings using scaleograms from vibration data of FEMTO bearings dataset.

Figures 5.7 and 5.8 show different scaleograms of Bearing1_1 horizontal and vertical vibration data (respectively) taken at different stages of the bearing's life.

Scaleograms can show important information about the bearing's health state, but different bearings can have different degradation patterns. For this reason, a dataset

Datasets	Operation conditions		
	Conditions 1	Conditions 2	Conditions 3
Learning set	Bearing1_1	Bearing2_1	Bearing3_1
	Bearing1_2	Bearing2_2	Bearing3_2
Testing set	Bearing1_3	Bearing2_3	
	Bearing1_4	Bearing2_4	
	Bearing1_5	Bearing2_5	Bearing3_3
	Bearing1_6	Bearing2_6	
	Bearing1_7	Bearing2_7	

Table 5.5: Different operating conditions and corresponding bearings in FEMTO dataset

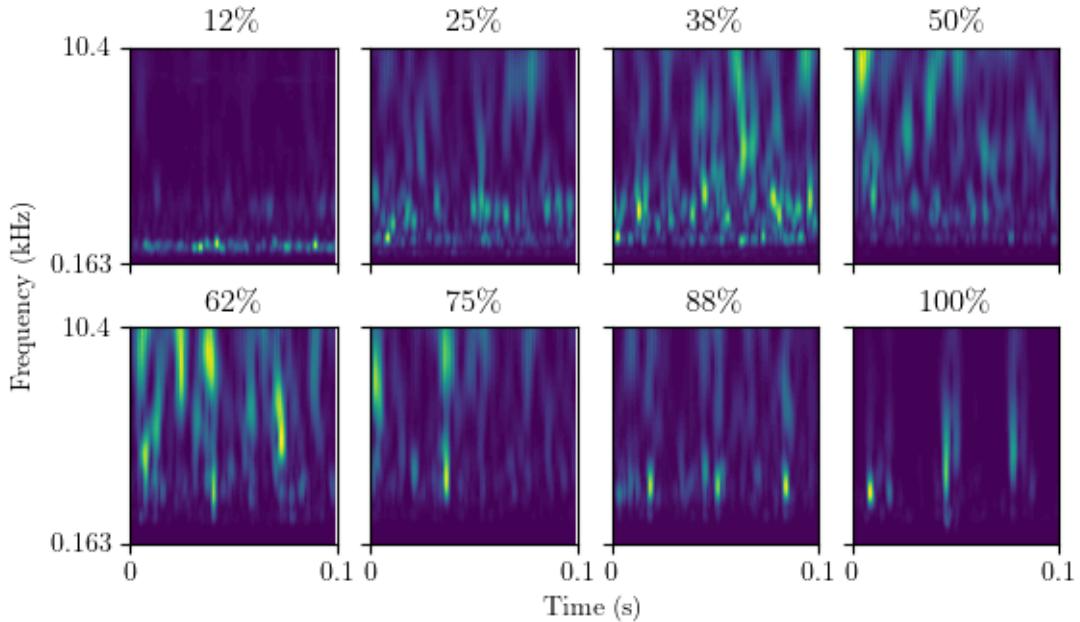


Figure 5.7: Scaleograms of different stages of Bearing1_1 life (horizontal vibrations)

of scaleograms extracted from vibration data is constructed. The first 80 vibration snapshots from each bearing are considered to represent a healthy bearing, while the last 80 vibration snapshots are considered faulty. This is considered as a binary classification task, where a convolutional neural network will be used to automatically extract features from scaleograms (both horizontal and vertical vibration data scaleograms) dataset and classify the bearing's health state. To avoid complexity and the cost of training a large model, the scaleograms are resized to a shape of (128×128) , so the input to the network is of shape $(2 \times 128 \times 128)$ where the two channels correspond to both horizontal and vertical vibration scaleograms. This data represents bearings from the first operation condition (from Bearing1_1 to Bearing1_7).

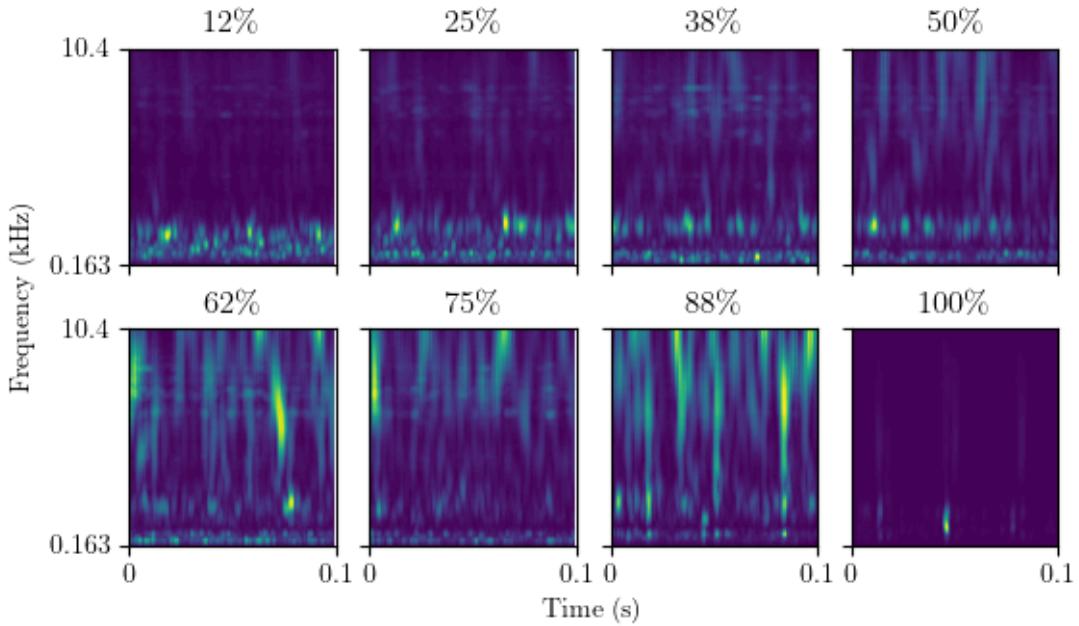


Figure 5.8: Scaleograms of different stages of Bearing1_1 life (vertical vibrations)

5.4.3 Detecting bearings failure using convolutional neural networks

Network architecture

For this task, a convolutional neural network (CNN) with 3 convolutional layers, 3 maxpooling layers and 3 fully-connected layers is used. The complete network architecture with the number of parameters in the network is represented in Table 5.6.

Training process

Generated dataset contained a total of 1120 samples, 716 samples were used for training, 180 for validation and 224 for test. The network was trained for 50 epochs with batch size of 64 samples. Figure 5.9 shows the training process.

Layer (type)	Output shape	Param #
Conv1 (Conv2D)	(None, 2, 128, 16)	18448
MaxPool1 (MaxPooling2D)	(None, 1, 64, 16)	0
Conv2 (Conv2D)	(None, 1, 64, 32)	4640
MaxPool2 (MaxPooling2D)	(None, 1, 32, 64)	0
Conv3 (Conv2D)	(None, 1, 32, 128)	18496
MaxPool3 (MaxPooling2D)	(None, 1, 16, 128)	0
Flatten1 (Flatten)	(None, 2048)	0
Dense1 (Dense)	(None, 1024)	524800
Dropout1 (Dropout)	(None, 512)	0
Dense2 (Dense)	(None, 64)	32832
Dropout2 (Dropout)	(None, 64)	0
Dense3 (Dense)	(None, 2)	66
<hr/>		
Total params: 559,346		
Trainable params: 599,346		
Non-trainable params: 0		

Table 5.6: Bearings health state classifier architecture

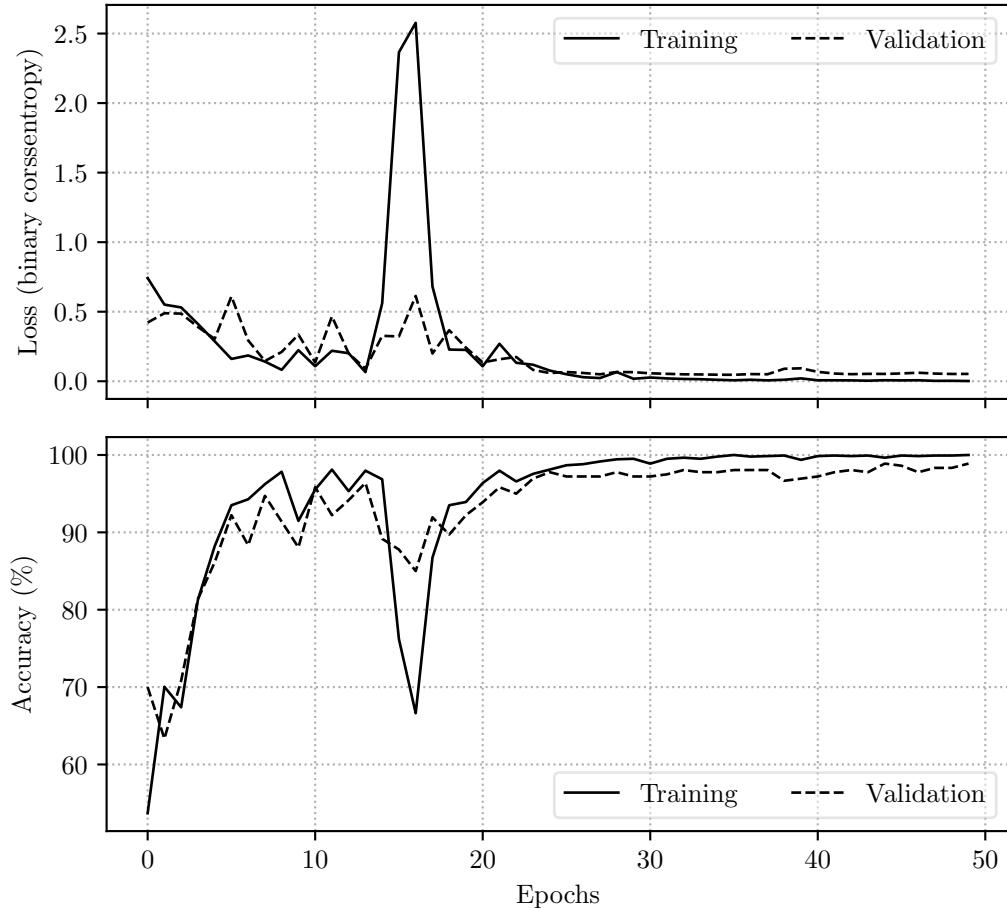


Figure 5.9: Bearings health state classifier training

Results discussion

The network achieved a perfect training accuracy of 100% and similar validation and test accuracy of around 98%. Table 5.7 shows loss and accuracy on train, validation and test data. Table 5.8 shows the additional precision, recall and F-1 score metrics.

	Loss	Precision
Train dataset	0.0017	100.00%
Validation dataset	0.0528	98.89%
Test dataset	0.0527	98.44%

Table 5.7: Training result

Metric	Value
Precision	0.9896
Recall	0.9694
F1 Score	0.9794

Table 5.8: Additional metrics for network performance

Figure 5.10 shows the ROC curve of the classifier:

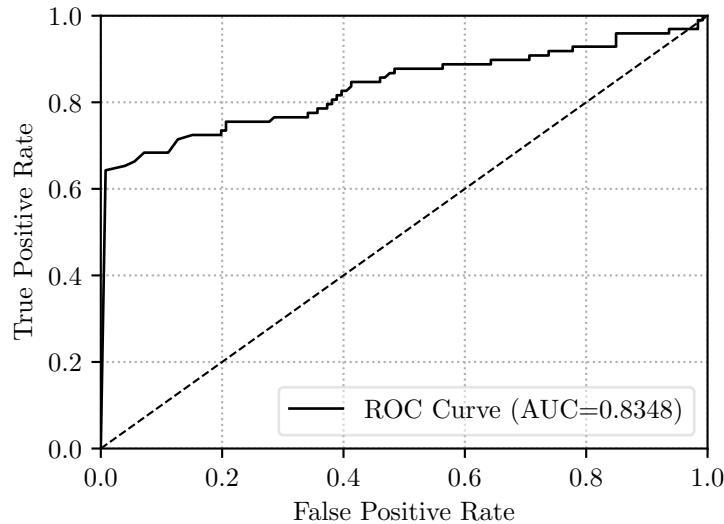


Figure 5.10: Bearings health state classifier ROC curve on test set

5.4.4 The need for new prognostics features

The prognostic model's input is vital to its predictions accuracy [33] and since that's the case, a feature selection step should be carried out while developing the model, which aims to select a set of appropriate features that can make the prediction more accurate [34]. That's why it's important to use concrete metrics which can quantify

the quality of our selected prognostics features and help comparing them directly and select the most suitable ones.

Trendability and monotonicity

In [33] the authors proposed a set of metrics which can be used to quantify the quality of prognostics features and directly compare their suitability for use in predictive models, two of these metrics are **monotonicity** and **trendability**. According to their definition, monotonicity refers to the nature of the variable whether it is increasing or decreasing and it has a value between 0 and 1 where a variable that is always increasing or decreasing will have monotonicity of 1. Monotonicity is defined by equation 5.2:

$$M = \frac{\text{no. of } \frac{d}{dx} > 0}{n - 1} - \frac{\text{no. of } \frac{d}{dx} < 0}{n - 1} \quad (5.2)$$

Trendability on the other hand was defined as whether a given feature follows the same trend (increasing or decreasing) across a population of systems. A feature that is always decreasing or increasing have a higher trendability than another one that doesn't follow a specific trend across different cases. Accordingly, trendability is defined by equation 5.3:

$$t_i = \frac{\text{no. of } \frac{d}{dx} > 0}{n - 1} + \frac{\text{no. of } \frac{d^2}{dx^2} > 0}{n - 2} \quad (5.3)$$

$$\text{Trendability} = 1 - \text{std}(t_i)$$

5.4.5 Trigonometric features and cumulative descriptors

In [8] the authors proposed a new feature extraction procedure from vibration data based on discrete wavelet transform (Section 2.2.6) and trigonometric functions. First, the raw vibration signal is decomposed using discrete wavelet transform with db4 wavelet and 4th level of decomposition, then the decomposition coefficients are scaled using trigonometric functions (e.g. asinh, atan), finally standard deviation of the scaled coefficients is calculated to obtain the final value. According to the authors, these features are less sensitive to noise and variability of raw vibration signals.

Table 5.9 shows mathematical definition of different classic and trigonometric features used in literature:

Figure 5.11 shows two trigonometric features (asinh and atan) for Bearing1_1 from FEMTO dataset. They have been filtered using Savitsky–Golay filter to reduce noise and variability.

Even that trigonometric features are less susceptible to noise compared to classical features (e.g. rms, skewness, kurtosis, etc.), that's not always the case. The authors also proposed the use of cumulative descriptors (i.e. running total) in order to assure monotonicity of the features. The cumulative feature associated with each feature is calculated according to equation 5.4 :

Trigonometric features	Formula
Standard deviation of asinh	$\sigma \left(\log \left[x_j + \sqrt{(x_j^2 + 1)} \right] \right)$
Standard deviation of atan	$\sigma \left(\frac{i}{2} \log \left(\frac{i+x_j}{i-x_j} \right) \right)$
Classic features	Formula
Entropy	$E(x) = \sum_j E(x_j)$
Energy	$e = \sum_{j=0}^n x_j^2$
Root mean square	$RMS = \sqrt{\frac{1}{n}(x_1^2 + \dots + x_n^2)}$
Skewness	$\frac{\sum_{j=1}^n (x_j - \bar{X})^3}{(n-1)\sigma^3}$
Kurtosis	$\frac{\sum_{j=1}^n (x_j - \bar{X})^4}{(n-1)\sigma^4}$
Upper bound	$\max(x) + \frac{1}{2} \frac{\max(X) - \min(X)}{n-1}$

Table 5.9: Prognostics trigonometric and classic features [8]

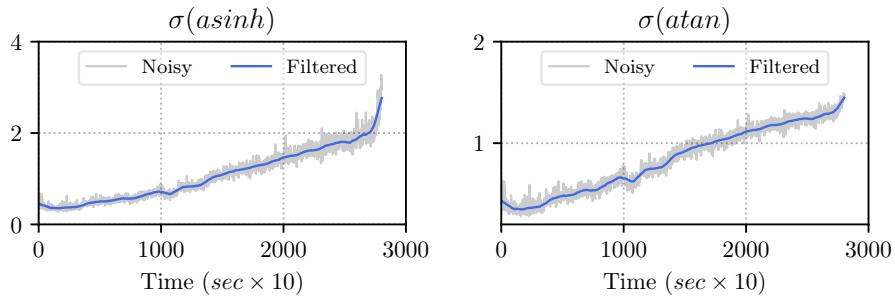


Figure 5.11: Trigonometric features of Bearing1_1

$$Cf_{nk} = \frac{\sum_{i=1}^n f_{ik}}{\sqrt{\text{abs}(\sum_{i=1}^n f_{ik})}} \quad (5.4)$$

In Figure 5.12 cumulative classic and trigonometric features are compared to their non-cumulative counterparts. Even with only visual inspection it is apparent that using cumulative descriptors greatly enhances the quality of prognostics features by reducing noise and fluctuations.

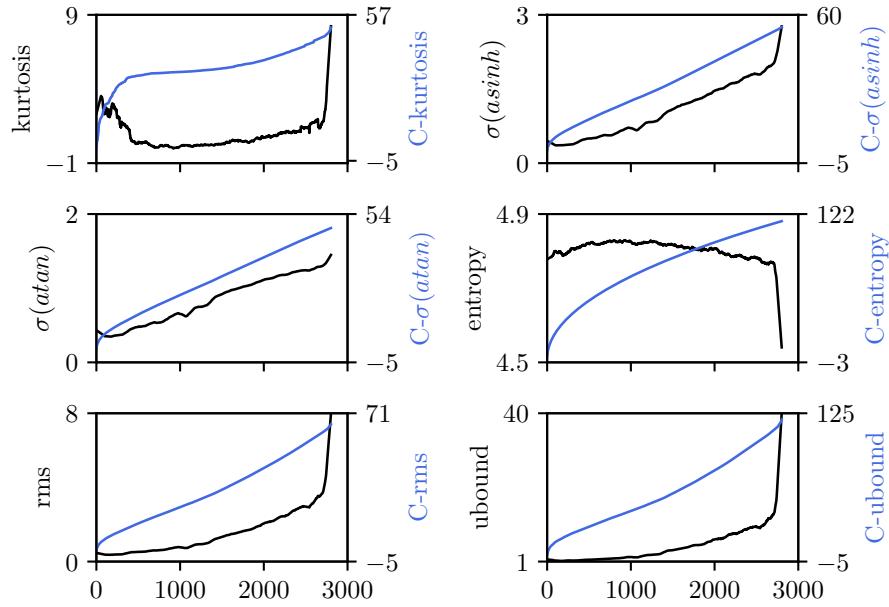


Figure 5.12: Classic and trigonometric cumulative features

To quantify the enhancement caused by using cumulative descriptors on prognostics features, monotonicity and trendability for each feature and its cumulative counterpart is calculated and summarized in Tables 5.10 and 5.11 respectively.

Feature	Monotonicity	Cumulative Feature	Monotonicity
$\sigma(\text{atan})$	0.486	$\text{C}-\sigma(\text{atan})$	1
$\sigma(\text{asinh})$	0.481	$\text{C}-\sigma(\text{asinh})$	1
kurtosis	0.059	C -kurtosis	0.998
entropy	0.035	C -entropy	1
rms	0.481	C -rms	1
ubound	0.287	C -ubound	1

Table 5.10: Monotonicity difference between trigonometric and classic features and their cumulative descriptors

Feature	Trendability	Cumulative Feature	Trendability
$\sigma(\text{atan})$	0.987	$\text{C}-\sigma(\text{atan})$	0.993
$\sigma(\text{asinh})$	0.989	$\text{C}-\sigma(\text{asinh})$	0.995
kurtosis	0.985	C -kurtosis	0.890
entropy	0.994	C -entropy	0.976
rms	0.988	C -rms	0.993
ubound	0.990	C -ubound	0.996

Table 5.11: Trendability difference between trigonometric and classic features and their cumulative descriptors

It is very obvious that features monotonicity increased significantly for cumulative features compared to non-cumulative ones. But for trendability the difference was less

significant. Figure 5.13 plots both monotonicity (x-axis) and trendability (y-axis) of both types of features:

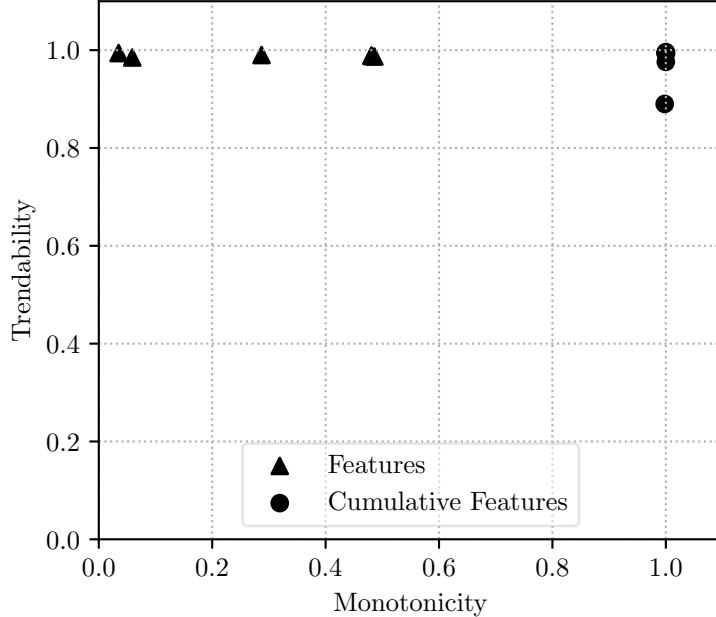


Figure 5.13: Features fitness

5.5 Application to oilfield equipment

5.5.1 Top Drive

In oil drilling rigs, Top Drive is a device that is used to turn the drill string and replace the conventional rotary table and kelly. The main advantage of Top Drives over conventional solutions is the ability to drill using three joints of pipes instead of just one, thus reducing drastically drilling time. Top Drives also helps drillers to minimize the cost and frequency of stuck pipe incidents [35].

Figure 5.14 shows Bentec 500-HT Top Drive.

From an engineering perspective, Top Drives are much more complicated systems than conventional rotary table and kelly, thus they need more rigorous maintenance programs to ensure their availability considering their major role in drilling operations. Every year the oil and gas technology is pushed harder, to drill in harder conditions and to drill deeper and more challenging oil wells, this poses stricter requirements on used equipment: to be able under and sustain harsher conditions while maximizing its availability which is of great importance in the field, for example Top Drives downtime cost can reach 1m\$ per day and cause significant delay to drilling operations [36].

In [37] the authors reported an industry survey that was conducted to assess the impression of different groups of people (operators, contractors, rig builders...) on the usage of Top Drives. The survey showed an average 60% satisfaction level across all the groups in the survey, which is an indicative that Top Drives aren't really meeting

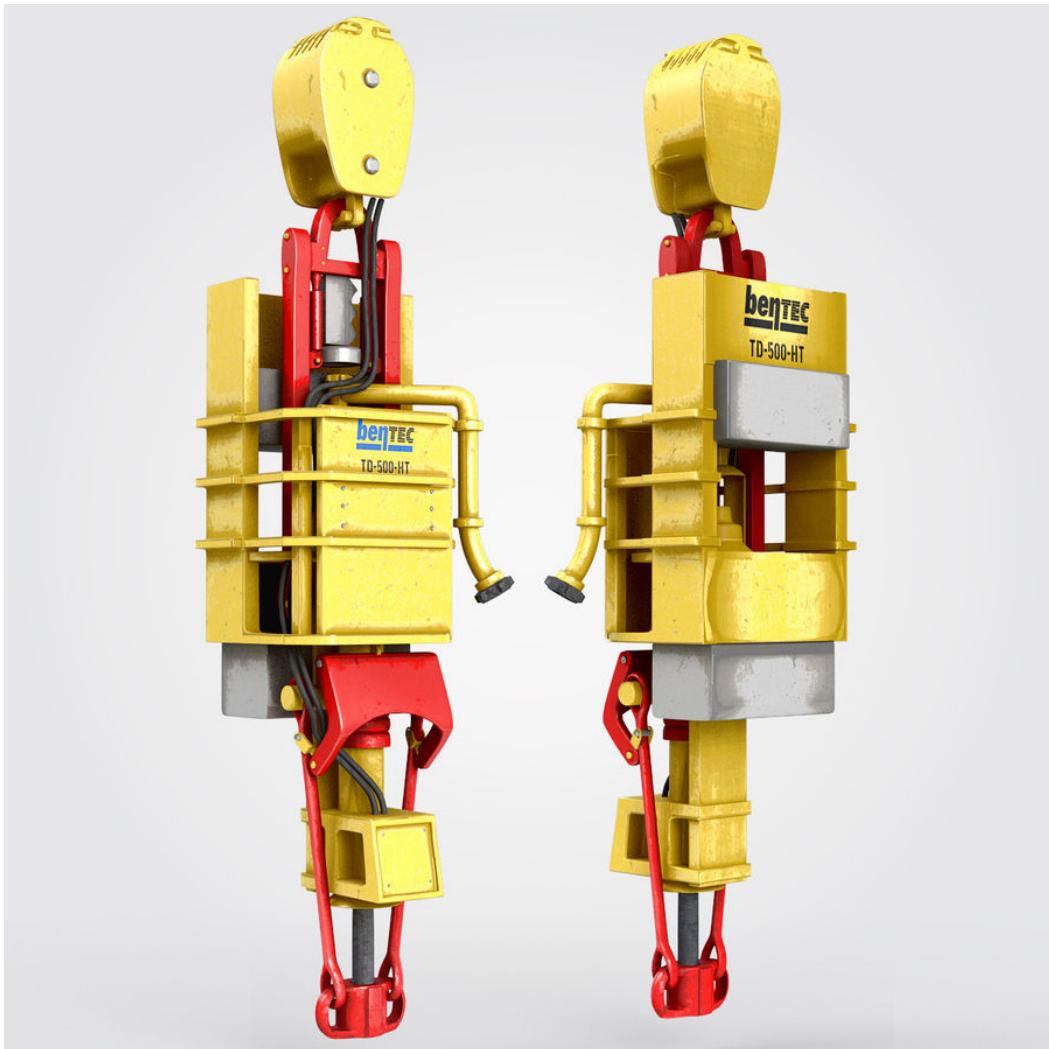


Figure 5.14: Topdrive Bentec 500-HT

the expectations in the industry. They were also asked which are the features they would like to see in Top Drives, the most wanted things are less downtime and better ability to detect failures before the breakdown of the system. This means that a more reliable preventive maintenance and prognostics approach is the most wanted improvement to these systems in the field.

5.5.2 Top Drive components

Top Drives are made of many sub-assemblies which are shown in Figure 5.15¹.

¹All the following figures of Top Drive components were taken from Bentec technical bulletin for TD-500-HT and TD-350-HT Top Drives

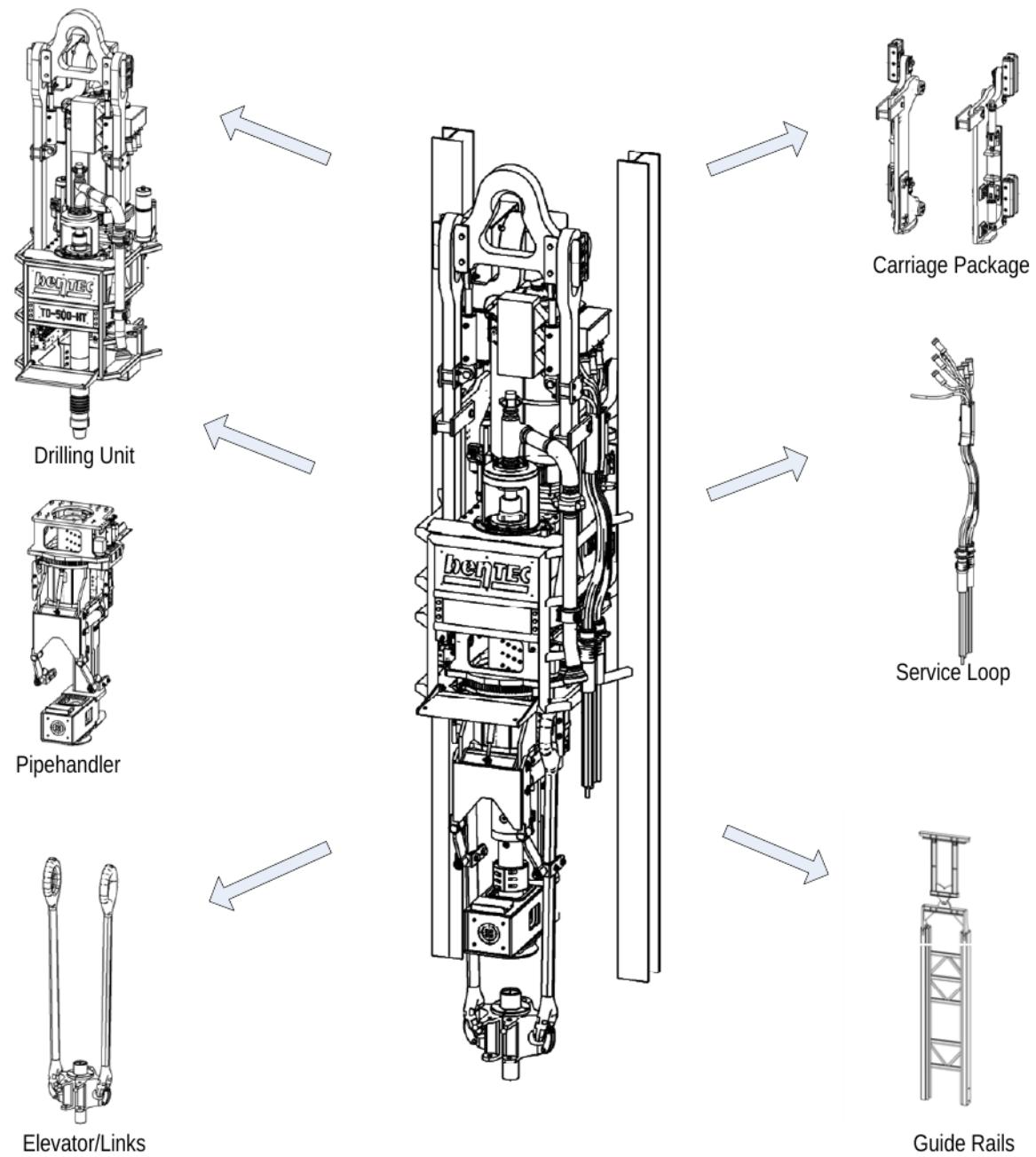


Figure 5.15: Bentec 500-HT Top Drive sub-assemblies

The Top Drive sub-assembly of interest here is the drilling unit (Figure 5.16) which is responsible for generating the rotation motion and transferring it to the drill string. Drilling unit is composed of protection frame, hydraulic unit, mud supply, hanger assembly and—the most important for the current discussion—a drive.

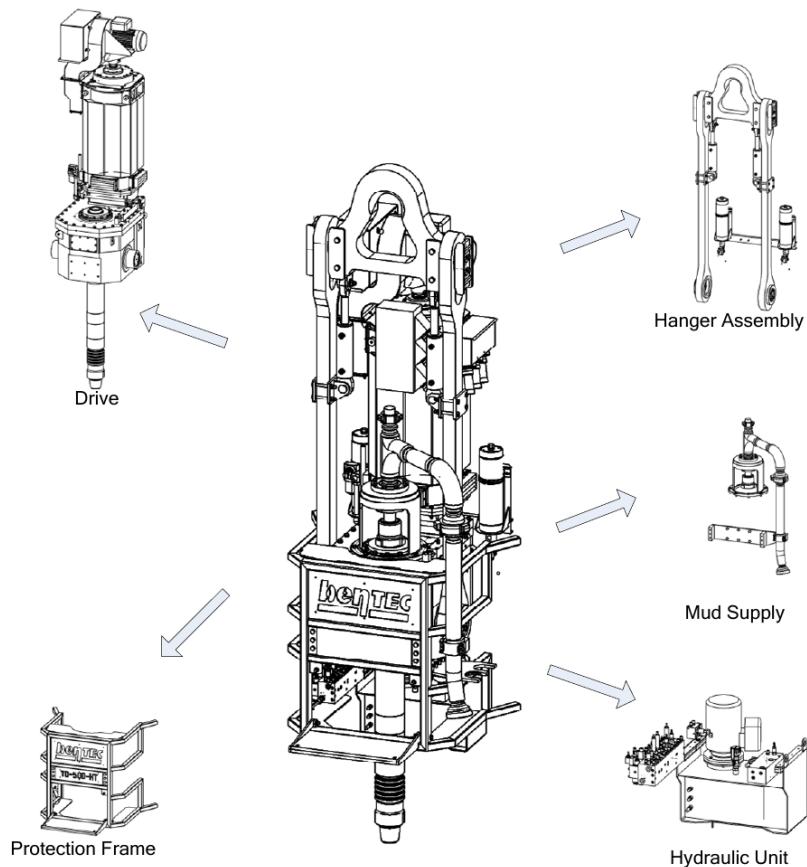


Figure 5.16: Bentec 500-HT Top Drive drilling unit

The drive itself is composed of:

- Engine cooling system
- Brake
- AC Motor
- Gearbox (Figure 5.17(a))
- Main shaft (Figure 5.17b)

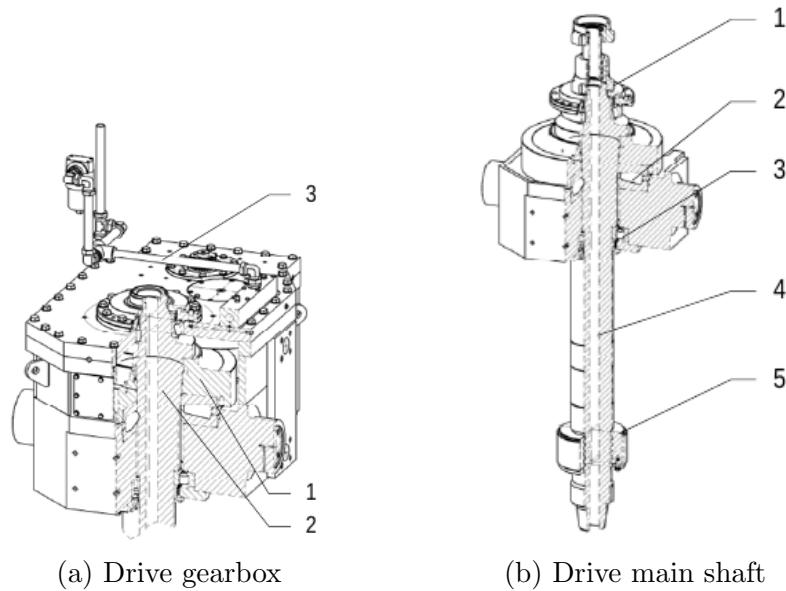


Figure 5.17: Main components of the drive

The gearbox has two stages with transmission ratio of 14:1. The speed of the engine is reduced twice then transferred to the bull-gear (1) of the gearbox. The bull-gear is located on the thrust bearing. Lubrication of the gearbox is done with combined splash/pressure lubrication (3). The main shaft (4) is powered by the bull-gear of the gearbox and located in the gearbox with the thrust bearing (2) and it is conducted via lower bearing (3). A wash pipe (1) is connected to execute drilling fluid. The main shaft also holds the load collar (5) which bears the link adapter to the drill pipe.

Figure 5.18 shows an example of a tapered bearing usually used in Top Drives. While Figure 5.19 shows the bearing within a Top Drive.

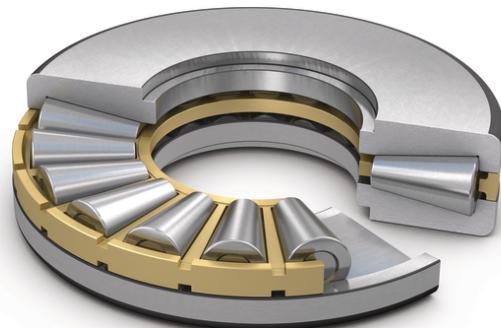


Figure 5.18: SKF tapered roller thrust bearing [6]

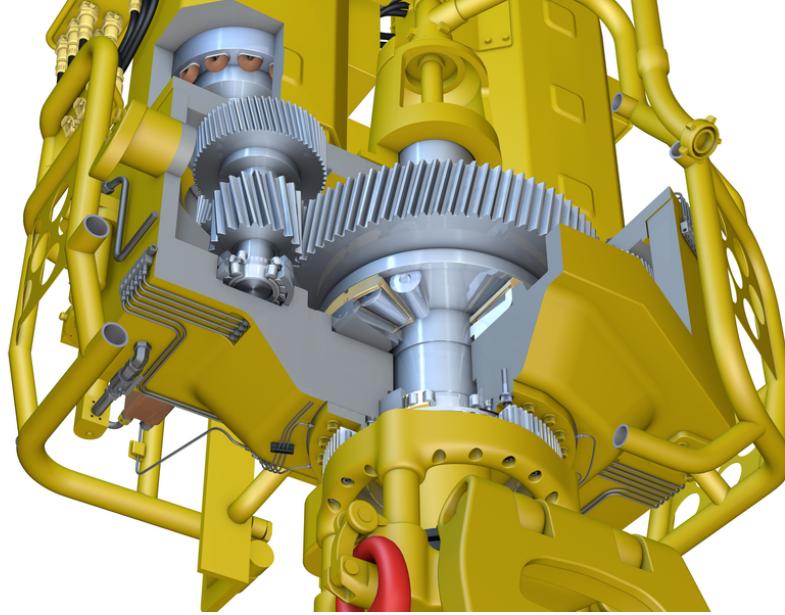


Figure 5.19: Bearing position within the Top Drive [7]

5.5.3 Proposed approach for Top Drive monitoring using neural networks

Top Drives are already equipped with many sensors to monitor their state like temperature, pressure and flow sensors, yet information that these sensors is only used in a simplified manner by operators [37]. In order to improve maintenance programs and adopt preventive maintenance and prognostics approaches, a more sophisticated and methodological approach to process data provided by these sensors need to be employed, along with introduction of new vibration sensors which are the main monitoring sensors used for bearings and gears, the elements which are present in the top drive and are work under high and varying loads.

Figure 5.20 is a simplified schematic representation of the proposed procedures to employ additional sensors (red dots) for vibrations in the drive to monitor different elements like gears and bearings. Enough sensors should be installed to capture vibrations in the three axes: vertical, horizontal and axial vibrations. When the Top Drive is in use, data collected from these sensors can be used to develop a neural network architecture to detect bearings degradation (Section 5.4) or—if enough historic data is available—classifying different degradation patterns (Section 5.3). Correctly detecting degradation or even classifying degradation patterns can be used to plan and prepare the appropriate maintenance actions and execute them in the appropriate time before any serious degradation happens which may cause serious non-productive time and high corrective maintenance costs.

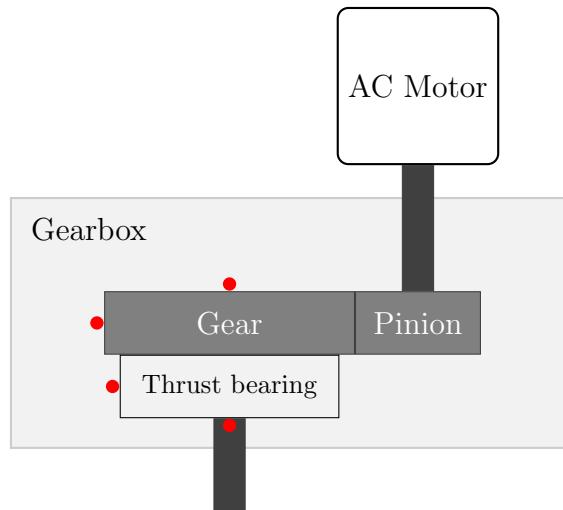


Figure 5.20: Proposed additional vibration sensors to monitor the drive

5.6 Conclusion

This chapter adopted a data generation approach from literature that converts vibration signals into images. Vibration signals used here corresponds to bearings with different types of faults and fault diameters. After converting raw signals into images, a CNN is used to classify transformed signals (i.e. images) into their corresponding faults types and diameters. The model achieves near-perfect classification accuracy on the test set.

Conclusion

Maintenance is an important process in all industrial facilities that ensures equipment availability and reduces equipment downtime. We can distinguish different types of maintenance actions where each type has a specific area of application. For critical equipment where repair cost is elevated and downtime can result in large losses, predictive maintenance was the preferred maintenance policy for many decades, yet with the technological advance in the last couple of decades where computational power become cheaper and sensors are ubiquitous, the industry started to shift towards what's called predictive maintenance and prognostics, which aim at providing predictions about the system's future behavior and potential breakdowns. There are many approaches to predictive maintenance and prognostics. This thesis focused on using data-driven methods, mainly machine learning and deep learning techniques and equipment monitoring data to develop predictive models that are able to learn degradation patterns of specific equipment and use it to predict the performance on new equipment. Chapter 4 used NASA C-MAPSS dataset to showcase the use of artificial neural networks (both fully-connected neural networks and LSTM networks) to model equipment degradation using data from sensors that monitor and measure different physical variables. The developed models showed great performance at predicting equipment remaining useful life (RUL). Later, the chapter presented the potential adoption of this approach in oilfields on oil equipment.

Chapter 5 focused on more specialized application of neural networks: monitoring bearings using vibration data for diagnostic and prognostic. The chapter used more sophisticated data processing techniques to extract suitable features from raw vibration data. Two different convolutional neural networks (CNN) were used, the first was used to diagnose bearings and classify fault type from vibration data, and the second used signal processing techniques and continuous wavelet transform to construct a dataset of scaleograms from vibration data which then was used as an input to the neural network to classify healthy and faulty bearings. The chapter also presented new feature extraction techniques from literature using trigonometric features and cumulative descriptors along with metrics used to quantify the suitability of these features to prognostics. Finally, a procedure was described to adopt solutions introduced in this chapter to Top Drives in oilfields.

Bibliography

- [1] K. Javed, R. Gouriveau, and N. Zerhouni, “State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels,” *Mechanical Systems and Signal Processing*, vol. 94, pp. 214–236, 09 2017.
- [2] F. Mangili, *Development of advanced computational methods for prognostics and health management in energy components and systems*. PhD thesis, Politecnico di Milano, Mar. 2013.
- [3] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *ArXiv e-prints*, mar 2016.
- [4] A. Saxena, K. Goebel, D. Simon, and N. Eklund, “Damage propagation modeling for aircraft engine run-to-failure simulation,” in *2008 International Conference on Prognostics and Health Management*, IEEE, oct 2008.
- [5] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varnier, “Pronostia: An experimental platform for bearings accelerated degradation tests,” pp. 1–8, 06 2012.
- [6] SKF, “Tapered roller thrust bearings.” Online.
- [7] SKF, “Skf helps develop 1000-ton top drive and consolidate oem supply chain.” Online.
- [8] K. Javed, R. Gouriveau, N. Zerhouni, and P. Nectoux, “A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling,” pp. 1–7, 06 2013.
- [9] ISO/IEC 2382:2015, “Information technology — vocabulary,” standard, International Organization for Standardization, Geneva, CH, 2015.
- [10] S. Kadry, ed., *Diagnostics and Prognostics of Engineering Systems*. IGI Global, 2013.
- [11] J.-H. Shin and H.-B. Jun, “On condition based maintenance policy,” *Journal of Computational Design and Engineering*, vol. 2, pp. 119–127, apr 2015.

BIBLIOGRAPHY

- [12] T. Tinga and R. Loendersloot, "Aligning phm, shm and cbm by understanding the physical system failure behaviour," in *European Conference of the Prognostics and Health Management Society 2014*, 07 2014.
- [13] E. R. Brown, N. N. McCollom, E.-E. Moore, and A. Hess, "Prognostics and health management a data-driven approach to supporting the f-35 lightning II," in *2007 IEEE Aerospace Conference*, IEEE, 2007.
- [14] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, "Prognostics and health management: A review on data driven approaches," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–17, 2015.
- [15] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, pp. 1483–1510, oct 2006.
- [16] M. I. M. Samuel Telford and I. Howard, "Condition based maintenance (cbm) in the oil and gas industry: An overview of methods and techniques," in *Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management*, 2011.
- [17] L. Eriksen, "The impact of digital on unplanned downtime." GE Digital, Oct. 2016.
- [18] A. Hess, J. S. Stecki, and S. D. Rudov-Clark, "The maintenance aware design environment: Development of an aerospace phm software tool," 2008.
- [19] A. Cubillo, S. Perinpanayagam, and M. Esperon-Miguez, "A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery," *Advances in Mechanical Engineering*, vol. 8, aug 2016.
- [20] 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, may 2018.
- [21] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation – a review on the statistical data driven approaches," *European Journal of Operational Research*, vol. 213, pp. 1–14, aug 2011.
- [22] R. Priemer, *Introductory Signal Processing*. World Scientific Publishing Co Pte Ltd, 1990.
- [23] Y. Lei, *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*, ch. Introduction and Background, p. 8. Elsevier - Health Sciences Division, 2016.
- [24] Y. Lei, *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*, ch. Introduction and Background, p. 9. Elsevier - Health Sciences Division, 2016.
- [25] Y. Lei, *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*, ch. Introduction and Background, p. 10. Elsevier - Health Sciences Division, 2016.

BIBLIOGRAPHY

- [26] Y. Chung, P. J. Haas, E. Upfal, and T. Kraska, “Unknown examples & machine learning model generalization,” *CoRR*, vol. abs/1808.08294, 2018.
- [27] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006. Christopher M. Bishop., Includes bibliographical references and index.
- [28] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, “The loss surfaces of multilayer networks,”
- [29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, oct 1986.
- [30] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, may 2015.
- [31] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, nov 1997.
- [32] L. Wen, X. Li, L. Gao, and Y. Zhang, “A new convolutional neural network-based data-driven fault diagnosis method,” *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 5990–5998, jul 2018.
- [33] J. Coble and J. Hines, “Identifying optimal prognostic parameters from data: A genetic algorithms approach,” *Annual Conference of the Prognostics and Health Management Society*, 01 2009.
- [34] K. Javed, R. Gouriveau, R. Zemouri, and N. Zerhouni, “Features selection procedure for prognostics: An approach based on predictability,” vol. 8, 08 2012.
- [35] Schlumberger, “Top drive.” Online Oilfield Glossary.
- [36] SKF, “Strength and reliability for the next generation of top drives.” Online.
- [37] P. Pournazari, D. Adams, P. Ashok, E. van Oort, and K. Holliday, “Real-time health monitoring of top drives for offshore operations using physics based models and new sensor technology,” in *SPE Deepwater Drilling and Completions Conference*, Society of Petroleum Engineers, 2016.