



HAL
open science

Stratégie d'implémentation de la maintenance prédictive dans un contexte d'industrie 4.0 : une approche basée sur le machine learning

Anthony Fombonne de Galathea

► To cite this version:

Anthony Fombonne de Galathea. Stratégie d'implémentation de la maintenance prédictive dans un contexte d'industrie 4.0 : une approche basée sur le machine learning. Automatic. Université de Bretagne Sud, 2023. English. NNT : 2023LORIS664 . tel-04522155

HAL Id: tel-04522155

<https://theses.hal.science/tel-04522155>

Submitted on 26 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE BRETAGNE SUD

ÉCOLE DOCTORALE N° 644

Mathématiques et Sciences et Technologies

de l'Information et de la Communication en Bretagne Océane

Spécialité : « *Automatique, Productique, Robotique* »

Par

« **Anthony FOMBONNE DE GALATHEAU** »

« **Predictive maintenance implementation strategy in an industry 4.0 context : a machine learning based approach** »

Thèse présentée et soutenue à « Lorient », le « date »

Unité de recherche : « Lab-STICC – CNRS UMR 6285 »

Thèse N° 664

Rapporteurs avant soutenance :

Zineb SIMEU-ABAZI Professeure, Polytech Grenoble

Philippe THOMAS Maître des conférences HDR, Université de Lorraine

Composition du Jury :

Président : Catherine DA CUNHA

Professeure, Ecole Centrale Nantes

Dir. de thèse : Nathalie JULIEN

Professeure, Université de Bretagne Sud (UBS)

Enc. de thèse : Alexandru-Liviu OLTEANU

Maître des conférences, Université de Bretagne Sud (UBS)

Invité(s) :

Steven LE GARREC PDG Dynavia

ACKNOWLEDGEMENT

I would like to express my eternal gratitude to Him, as I could not have achieved this work.

This thesis was carried out by the company Dynavia with the collaboration of the Laboratory of Sciences and Techniques of Information, Communication and Knowledge (Lab-STICC) at the University of Southern Brittany (UBS), more particularly at the National Engineering School of Southern Brittany (ENSIBS), for which I thank Mr. Le GARREC Steven and Pr. MARTIN Éric.

It is with great pleasure that I express here my deep gratitude to my thesis director Professor JULIEN Nathalie and my supervisor Dr. OLTEANU Alexandru-Liviu, for their supervision, their support during all these years of thesis. I am grateful for the time and energy you devoted to my thesis. Your expertise, advice and commitment have been invaluable in the realisation of this research work. Thank you for the positive impact you have had on my personal and scientific development.

I would like to express my sincere thanks to the company Dynavia, represented by Mr. LE GARREC Steven, for the confidence that was given to me, the means put in place, the availability to carry out this work.

I would like to express my gratitude to the members of the Jury for having accepted to follow this work, your expertise is invaluable and allowed me to improve the quality of this manuscript.

I would also like to thank the members of the individual thesis monitoring committee, Pr. PENZ Bernard and Pr. TAMZALIT Dalila, for their expertise, their advice, their availability and their listening skills.

I would like to thank my research laboratory colleagues Dr. KERNACH BEY Mohamed, Dr. TOUBLANC Thomas, Dr. ALEM Salwa, Dr. HAMZAOUI Mohamed Adel and many others for their good humour, advice and listening.

It is with great emotion that I thank Dr. BEN DAHOU Dounia, my dear wife, for all these years of sacrifice and patience. She has always supported, encouraged and advised me so that I could go through with this project. Her scientific advice was precious and contributed to the writing of this manuscript. These words will never be strong enough to thank her properly. I also thank Dr. BEN DAHOU Abdelkader, my brother-in-law, for his advice throughout the thesis, whether personal or scientific. I was very lucky to have you both.

Finally, I would like to thank from the bottom of my heart all the members of my family in France and in Morocco, especially my parents, my sisters, my brother, my grandmother, and all my family-in-law for their support and encouragement. Your presence has allowed me to remain strong in moments of doubt and to remain humble in moments of happiness.

TABLE OF CONTENTS

List of Figures	13
List of Tables	14
Acronyms	15
General Introduction	17
Thesis objectives	19
Organisation of the thesis	19
Publication related to the thesis	20
1 From failure-based maintenance to predictive maintenance	21
1.1 Introduction	21
1.2 Concepts and definitions	21
1.3 The evolution of maintenance	22
1.3.1 Industry 1.0	23
1.3.2 Industry 2.0	23
1.3.3 Industry 3.0	23
1.3.4 Industry 4.0	24
1.3.5 Maintenance objectives	25
1.3.6 Different types of maintenance	26
1.3.6.1 Corrective maintenance	26
1.3.6.2 Preventive maintenance	27
1.3.6.3 Improvement maintenance	28
1.3.6.4 Predictive maintenance	28
1.4 The state of data in maintenance	31
1.4.1 Data in Maintenance 1.0	31
1.4.2 Data in Maintenance 2.0	32
1.4.3 Data in Maintenance 3.0	32
1.4.4 Data in Maintenance 4.0	32

TABLE OF CONTENTS

1.5	The effectiveness of data-driven predictive maintenance	33
1.5.1	Logistic regression	34
1.5.2	Support Vector Machine	36
1.5.3	Decision Tree and Random Forest	37
1.5.4	Artificial Neural Network	37
1.5.5	Deep Neural Network	38
1.5.6	Auto-Encoder	39
1.5.7	Summary table	40
1.6	Conclusion	42
2	Anomaly detection in times series	43
2.1	Introduction	43
2.2	Tools and methods	43
2.2.1	Programming libraries for data science	45
2.2.2	Cloud computing	46
2.2.3	Machine Learning Methods	47
2.2.3.1	Regression	48
2.2.3.2	Classification	49
2.2.3.3	Statistical techniques	50
2.2.3.4	Clustering techniques	50
2.2.3.5	Expert systems	51
2.2.3.6	Pattern extraction	51
2.2.3.7	Spectral analysis	51
2.2.4	Deep Learning Methods	51
2.2.4.1	Artificial Neural Network (ANN)	51
2.2.4.2	Recurrent Neural Network (RNN)	55
2.2.4.3	Long Short-Therm Memory (LSTM)	56
2.2.4.4	Gated recurrent Unit (GRU)	58
2.2.5	Evaluation metrics	59
2.2.5.1	Confusion matrix	59
2.2.5.2	Regression metrics	60
2.3	Anomalies detection in times series	61
2.3.1	Fields of application	62
2.3.2	Types of anomalies	63

2.3.2.1	Point anomalies	63
2.3.2.2	Contextual anomalies	63
2.3.2.3	Collective anomalies	63
2.3.3	Machine learning-based anomaly detection	64
2.3.3.1	Regression	64
2.3.3.2	Classification	65
2.3.3.3	Statistical techniques	66
2.3.3.4	Clustering techniques	66
2.3.3.5	Expert systems	67
2.3.3.6	Pattern extraction	67
2.3.3.7	Spectral analysis	68
2.3.4	Deep learning-based anomaly detection	68
2.3.4.1	Artificial Neural Network (ANN)	68
2.3.4.2	Long Short-Therm Memory (LSTM)	69
2.3.4.3	Gated Recurrent Unit (GRU)	71
2.4	Conclusion	72
3	Predictive maintenance strategy applied in an industrial context	73
3.1	Introduction	73
3.2	Understanding and identifying the study	75
3.2.1	Functional analysis	75
3.2.2	Early planning for confidentiality and access to data in research studies	78
3.2.3	Overall Equipment Effectiveness (OEE)	80
3.2.4	Unavailability per groups and Pareto diagram	82
3.2.4.1	Unavailability per groups	82
3.2.4.2	Pareto diagram	89
3.2.5	Failure rate and maintenance history	92
3.2.5.1	Failure rate	92
3.2.5.2	Calculation of the system reliability	93
3.2.5.3	Application to our case study	95
3.2.5.4	Maintenance history	98
3.3	Data collection and preparation	99
3.3.1	Mapping the data sources	99

TABLE OF CONTENTS

3.3.2	Data extraction	99
3.3.3	Preparation	100
3.3.4	Modeling	100
3.3.5	Evaluate	101
3.3.5.1	Setting up a machine learning process	101
3.3.5.2	Maintainability process of the machine learning model	101
3.3.6	Maintenance decisions and actions	102
3.3.6.1	Presentation of prediction results and KPIs	102
3.3.6.2	Reflection	102
3.3.6.3	Back to the design	103
3.4	Conclusion	103
4	Anomaly detection based on machine learning	105
4.1	Introduction	105
4.2	Industrial Use-case	106
4.2.1	Case study: Line loading problem	111
4.2.2	Case study: Overpressure problem and leak detection	113
4.3	Methodology	114
4.3.1	Upload dataset	115
4.3.2	Preprocessing	117
4.3.3	Standardization	118
4.3.4	Split dataset	119
4.3.5	Data preparation for an LSTM network	119
4.3.6	Description of the LSTM model architecture	121
4.4	Experimentation	122
4.4.1	Results	122
4.4.2	Experimentations: LSTM with different configurations	125
4.4.3	Experimentations: GRU with different configurations	126
4.4.4	Experimentations: LSTM with pressure downstream	128
4.5	Comparison of results	130
4.6	Conclusion	131
	Conclusion and perspectives	133
	Conclusion	133
	Perspectives and future works	134

Bibliography	137
Appendix	151

LIST OF FIGURES

1	Synoptic of the Dynavia machine.	18
1.1	Industrial and maintenance revolutions	22
1.2	The four objectives of maintenance [Yves, 1992]	25
1.3	Maintenance Types	26
1.4	Predictive maintenance in a context of P-F curve adapted from [Bousdekis and Mentzas, 2021]	29
1.5	Comparison of RM, PM and PdM on the cost and frequency of maintenance work from [Ran et al., 2019]	30
1.6	Comparison of maintenance data in different maintenance ages, inspired from [Jasiulewicz-Kaczmarek et al., 2020]	32
2.1	Comparison between biological and artificial neuron from [Mwandau and Nyanchama, 2018]	52
2.2	Example of a multilayers perceptron	54
2.3	Gradient descent	55
2.4	LSTM Cell	56
2.5	Forget Cell	57
2.6	Input Cell	57
2.7	State Cell	58
2.8	Output Cell	58
2.9	GRU Unit	59
2.10	Example of a point anomaly in a time series	63
2.11	Contextual anomaly in a monthly temperature time series (Chandola et al., 2009)	64
2.12	Example of collective anomaly in time series	64
2.13	Global deep learning techniques for anomaly detection in times series	65
3.1	Predictive maintenance strategy workflow inspired from [Achouch et al., 2022]	73

LIST OF FIGURES

3.2	Analysis funnel to identify a candidate for predictive maintenance	75
3.3	SADT Black box	76
3.4	Example of a SADT model by level	77
3.5	Extract from a Dynavia function block assembly	79
3.6	OEE from ISO 22400-2 standard	80
3.7	The different stages of maintenance	83
3.8	Extract of alarms for analysis of the phenomenon of overlapping alarms . .	87
3.9	Timeline of an overlapping alarm phenomenon	88
3.10	Principe de Pareto - 80/20	89
3.11	Pareto chart DYN00A from maintenance history	92
3.12	RBD representation of elements A and B in series	93
3.13	RBD representation of elements A and B in parallel	94
3.14	RBD representation from the functional analysis	94
3.15	Extract failure rate - OREDA injection pump - page 187 - [Handbook, 2002]	96
3.16	Extract failure rate - OREDA filter - page 811 - [Handbook, 2002]	96
3.17	Extract failure rate - OREDA ball valve - page 576 - [Handbook, 2002] . .	97
3.18	Block diagram of the reliability of storage and dosing of formulated prod- ucts from the BDF representation 3.14.	98
3.19	Maintainability process of the machine learning model	102
4.1	Process Dynavia	106
4.2	An extract from the FP rack	107
4.3	Piping and Instrumentation Diagram (PID) simplified - Dosing pump . . .	108
4.4	Piping and Instrumentation Diagram (PID) simplified - Dosing pump . . .	110
4.5	Case 1 - Correct Line loading	111
4.6	Case 1 - Line loading problem	112
4.7	Case 2 - Evolution of overpressures and leaks	113
4.8	Process machine learning	114
4.9	Trendviewer - Data identification	115
4.10	Transfer of customer data to a NBaaS cloud solution	116
4.11	Example Dataset - Density in XML format	116
4.12	Example Dataset - Density in CSV format	117
4.13	Example Dataframe - Density	117
4.14	Example Dataframe	120
4.15	LSTM model for one variable	121

4.16 Original and reconstructed density values over the test data - Line E . . . 123

4.17 Estimated dosing pump anomalies - Line E 123

4.18 Original and reconstructed pressure values over the test data 129

4.19 Anomaly detection on the original signal - Line E 130

4.20 All experiments on anomaly detection based on LSTMs and GRUs archi-
 tectures. 152

LIST OF TABLES

1.1	Advantages and disadvantages of different maintenance techniques - Adapted from [Abbasi et al., 2018] and [Shetty, 2018]	31
1.2	Summary of the data-driven predictive maintenance method using three types of machine learning algorithms from [Zhang et al., 2019b]	40
2.1	A comparison between Python and R	45
2.2	Overview of the different articles on anomaly detection with LSTM	71
2.3	Overview of the different articles on anomaly detection with LSTM	72
3.1	Extract alarm classification - Dynavia	86
3.2	Unavailability by group in hours - Dynavia	88
3.3	Table for plotting Pareto's law - Dynavia case	91
4.1	Available data	109
4.2	Maintenance tickets received	111
4.3	Comparison of dataset dimensions before and after preprocessing	118
4.4	Grid search parameter values	122
4.5	Estimated dosing pump anomalies - Line E	124
4.6	Model results and evaluations by changing the parameter units of the network a layer LSTM with a time steps of 50	125
4.7	Model results and evaluations by changing the parameter units of the network a layer LSTM with a time steps of 20	125
4.8	Model results and evaluations by changing the parameter units of the network a layer GRU with a time steps of 50	126
4.9	Model results and evaluations by changing the parameter units of the network a layer GRU with a time steps of 20	127
4.10	Model parameters with the pressure variable	128
4.11	Metrics - LSTM model for pressure downstream	129
4.12	Metrics - LSTM model for pressure downstream	130
4.13	Comparison results with density and vibration data using LSTM model.	131

ACRONYMS

- AE** Auto-Encoder. 39
- ANN** Artificial Neural Network. 34
- ARIMA** AutoRegressive Integrated Moving Average. 33
- ARMA** AutoRegressive Moving Average. 36
- CCNN** Cascade-Correlation Neural Network. 35
- CMMS** Computerized Maintenance Management System. 24, 32
- DCS** Distributed Control System. 86
- DNN** Deep Neural Network. 39
- DT** Decision Tree. 37
- DTr** Density Transmitter. 108
- ERP** Enterprise Resource Planning. 24
- FNN** Feedforward Neural Network. 38
- IBC** Intermediate Bulk Container. 108
- IIOT** Industrial Internet of Things. 32
- IPC** Industrial Personal Compute. 33
- LR** Logistic Regression. 34
- LT** Loading Time. 81
- LTr** Level Transmitter. 108
- MLR** Multiple Logistic Regression. 34
- MQTT** Message Queuing Telemetry Transport. 33
- OEE** Overall equipment effectiveness. 7, 74, 80

OPT Operating Time. 81

PID Piping and Instrumentation Diagram. 12, 108, 110

PLC Programmable Logic Controller. 33

PTr Pressure Transmitter. 108

RBD Reliability Block Diagrams. 92

RF Random Forest. 37

RUL Remaining Useful Lifetime. 35

RVM Relevance Vector Machine. 34

SADT Structured Analysis and Design Technique. 76

SCADA Supervisory Control And Data Acquisition. 86

SVM Support Vector Machine. 33, 34

TPM Total Productive Maintenance. 24

WPD-PCA Wavelet Packet Decomposition-Principal Component Analysis. 38

GENERAL INTRODUCTION

With the world's growing population, constant demand for food and limitations on the expansion of agricultural land, scientists are looking to innovate more efficient methods to improve crop production [Hoseini et al., 2013].

Industry 4.0, also known as the fourth industrial revolution, has brought significant technological advances in many sectors including agriculture. There are many applications in seed processing, such as automation, real-time data collection and analysis that allow sorting of good or bad seeds by optical treatment, all of which offer new opportunities to improve the efficiency and quality of seed processing. Seed treatment is a key element of the modern agricultural industry, aimed at ensuring the quality and performance of the seed used for crop production. Seed treatment refers to the application of certain physical, chemical or biological agents to the seed prior to sowing in order to suppress, control or repel pathogens, insects and other pests that attack seeds, seedlings or plants and it ranges from a basic dressing to coating and pelleting [Sharma et al., 2015].

It is in this context that the Dynavia system is positioned as a major player with state-of-the-art components, offering innovative solutions for the effective treatment of seeds while respecting the safety of operators in the face of the danger of phytosanitary products.

Dynavia is a start-up that was founded in 2013 in Pontivy, France, and is the result of a merger of several companies, including Le Garrec and Stolz. Le Garrec is a company specialising in the development of fluid processes for the food, animal nutrition, environment and petrochemical sectors. Stolz is a company that designs, manufactures and installs equipment that enables industrialists to handle and transform powdery or granular products.

The Dynavia machine, represented in the Figure 1, is composed of several functions, including Seeds, Formulated Products, Clean Water, Powder, Premix, ATM, Dirty Water and Bagging. The Seeds function allows the dosing of seeds for each treatment, with the

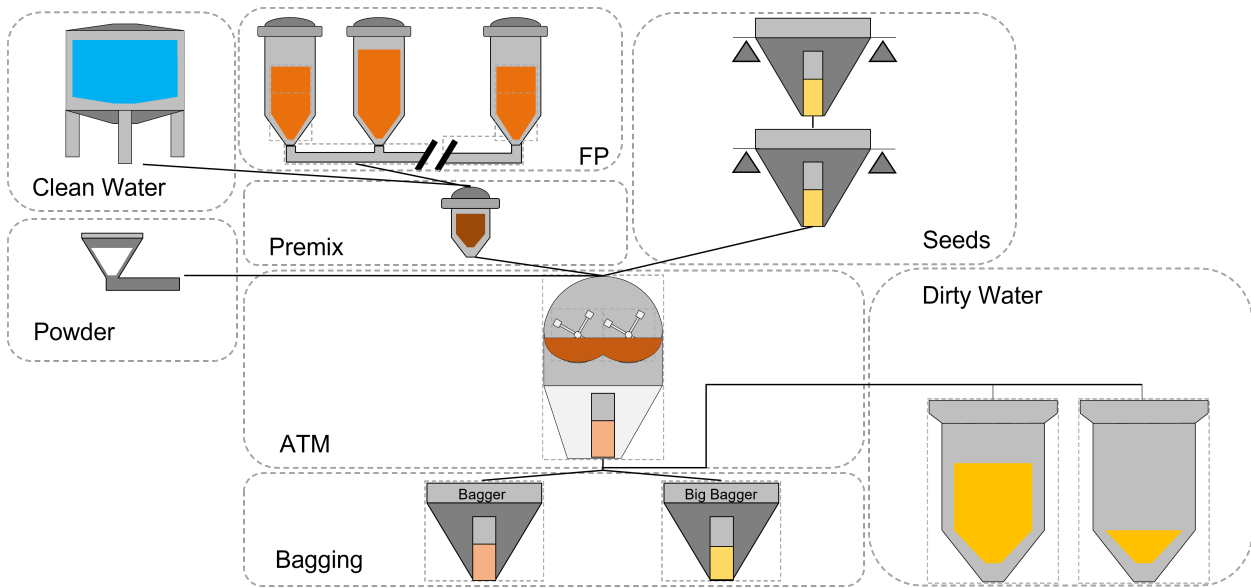


Figure 1 – Synoptic of the Dynavia machine.

upper hopper storing the seeds for the first time and then pouring them into the lower hopper for accurate dosing of the next treatment. The Formulated Products function involves the storage of the formulated products in tanks, with dosing pumps taking the necessary quantity of product corresponding to the quantity of seeds. Clean Water is used to store hot water and distribute it throughout the installation, which is mainly used for cleaning. The Powder function allows the dosing of powder for each seed treatment. The Formulated Products are then incorporated into a tank, mixed and pressurised to be injected into the ATM. The seeds fall into the ATM and are agitated, with nozzles injecting the mixture of formulated products and powders onto the seeds during the process, which will eventually be dried with dehydrated air. The Dirty Water area allows the recovery of waste water from an insecticide or fungicide treatment during one of the Dynavia’s washes. The water is then evacuated for destruction or reused for the same recipe. Finally, the Bagging function completes the seed treatment, with the ATM hatches opening and the seeds being bagged in bags or being placed in a large bin to be sold.

Once the Dynavia system is purchased, the customer can opt for an after-sales service contract to ensure continuous availability of their production. This means that if problems arise, the customer can contact Dynavia’s maintenance department to resolve the issues. However, this type of maintenance, known as corrective maintenance, only takes place when a failure is detected. Over time, the economic losses gradually increase, which

is not ideal for the customer. Some customers may experience a loss of more than €20,000 per hour of downtime.

Dynavia has nine machines sold throughout France, eight of which are fully digital, each producing over 60,000 different variables. Depending on the version, a machine can be equipped with 200-300 sensors. The maximum output of a machine is 40 tonnes/hour. The main processing tower is more than 20 meters high and to move the whole system, 6 semi-trailers are needed. An overview is given in chapter 4.

Thesis objectives

It is in this real interest that the company Dynavia wished to set up research work on predictive maintenance. The aim of this thesis work can be distinguished in two ways:

- To establish a generic predictive maintenance strategy that can be put into practice in any industrial case.
- Once the component has been identified, apply the predictive maintenance strategy, with the associated methods and tools, notably machine learning or deep learning.

Organisation of the thesis

In view of the objectives of the thesis and the work carried out, we have chosen to structure this manuscript around four chapters.

- The first chapter is a bibliographic synthesis on failure-based maintenance towards predictive maintenance. This chapter also gives a global vision of the evolution of maintenance through the four industrial revolutions with definitions, objectives, types, state of the art data and efficiency of data-based maintenance.
- In a second chapter, the different techniques and methods associated with the problem of anomaly detection in time series are presented. Then, we detail the various types of anomalies treated in the literature with the context of use and the methods of evaluation.
- The third chapter will be devoted to the proposal of a predictive maintenance strategy in an industrial case. This proposal is based on a set of methods and analysis to

identify an ideal component for predictive maintenance according to several criteria and then it continues with the implementation of a machine learning process to create the model, evaluate it, maintain it and finally make maintenance decisions.

- The fourth and last chapter of the manuscript deals with anomaly detection in a real case. The previous strategy that allowed us to identify a functionality of the equipment to study and more precisely a dosing pump for a given customer. Through chapter 2, we apply the tools and methods to perform our experiments.

Finally, a general conclusion deals with the main results obtained from this thesis work, as well as the perspectives that we think are relevant.

Publication related to the thesis

The present study represents the written work that was submitted and presented at an international conference:

- DE GALATHEAU, Anthony Fombonne, OLTEANU, Alexandru-Liviu, JULIEN, Nathalie, et al. Long Short Term Memory-based anomaly detection applied to an industrial dosing pump. IFAC-PapersOnLine, 2022, vol. 55, no 2, p. 240-245.

This article corresponds to the contribution presented in chapter 4. It presents the application of an LSTM deep learning method, to be optimised, to perform anomaly detection on an industrial case: a dosing pump.

FROM FAILURE-BASED MAINTENANCE TO PREDICTIVE MAINTENANCE

1.1 Introduction

In this chapter of the state of the art, we introduce failure-based maintenance to predictive maintenance. We present the concepts and definitions related to maintenance, its evolution through the industrial revolutions, its objectives, and the existing types. We then detail several works illustrating predictive maintenance based on the evolution of numerical data and its effectiveness using machine learning and deep learning techniques.

1.2 Concepts and definitions

To understand the definition of maintenance in detail, [Laloux, 2009] proposes a set of definitions according to several standards:

Definition 1.2.1 *The term "maintenance" defined by NF EN 13306 (June 2001) is: all the technical, administrative and management actions during the life cycle of an asset, intended to maintain it or restore it to a state in which it can perform the required function.*

Definition 1.2.2 *The term "maintenance management" by FD X60-000 (May 2002): all the activities of the management bodies which determine the objectives, strategy and responsibilities concerning maintenance and which implement them by means such as the planning, control and monitoring of maintenance and the improvement of methods in the company, including the economic aspects.*

Definition 1.2.3 *The term "required function" by NF EN 13306 (June 2001): function or set of functions of an asset considered necessary to provide a given service.*

Definition 1.2.4 The term "asset" by FD X60-000 (May 2002): any element, component, mechanism, subsystem, functional unit, equipment or system that can be considered individually.

1.3 The evolution of maintenance

The intelligence of the human being make them an exceptional creature. Throughout history, humans have utilized this quality to innovate and find ways to minimize effort. They will persist in leveraging their imagination and intelligence to develop systems that can handle monotonous and arduous tasks, while remaining under human supervision. Figure 1.1 briefly depicts the industrial revolutions that shifted from artisanal, agrarian societies to commercial and industrial societies.

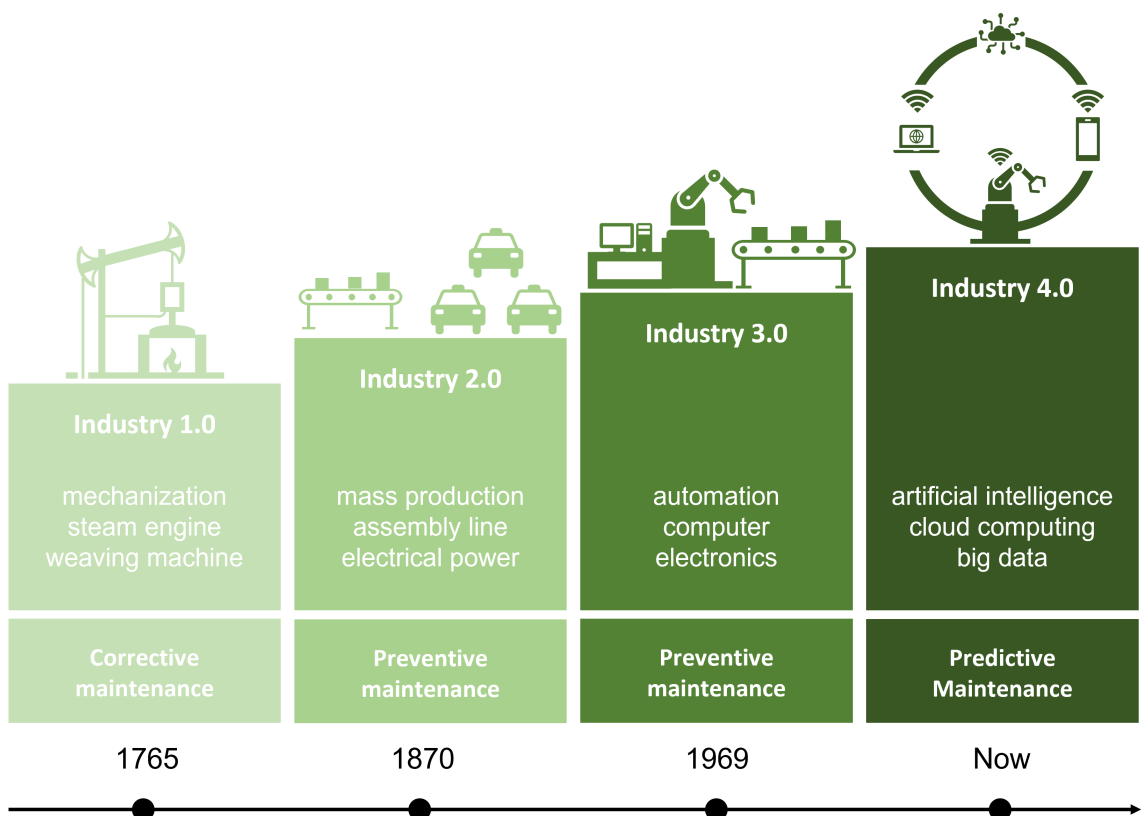


Figure 1.1 – Industrial and maintenance revolutions

1.3.1 Industry 1.0

This first revolution began in the 18th century in the United Kingdom before spreading to the rest of Europe, then the United States and Japan. It was marked by several technological developments such as:

- The steam engine with condensation chamber, created by James Watt in 1769.
- The first automatic weaving machine, created by Richard Arkwright in 1779.
- The first steam locomotive appeared in 1804.

Thus, this period increased the production of textiles, iron and steel by transforming production from manual to partially mechanised. This period brought about a strong demographic growth, migration, economic upheaval with the concentration of wealth to factory owners and politicians [Julien and Martin, 2021] and [Tahiri, 2021].

1.3.2 Industry 2.0

This second industrial revolution began at the end of the 19th century. It was based on the extraction of oil and gas, the invention of electricity and the internal combustion engine. These innovations led to the creation of new industrial sectors such as the automobile, aviation, chemicals and electricity production, and means of communication such as Graham Bell's telegraph and telephone in 1876. The engineer Frederick Winslow Taylor introduced Taylorism in 1911, a scientific organisation of work to increase the productivity of employees and the industrialist Henry Ford developed mass production with assembly lines in the 1910s.

During the first two industrial revolutions until the beginning of 1900, maintenance interventions were corrective. They consisted of cleaning, lubrication and waiting for the appearance of failures leading to breakdowns before attempting to repair and replace damaged parts [Julien and Martin, 2021] and [Tahiri, 2021].

1.3.3 Industry 3.0

This third industrial revolution began in the United States, then in Japan and the European Union in the 1960s and 1970s with the emergence of electronics, the creation of the transistor, the development of telecommunications, computers and nuclear power. Essentially, two products had an impact on industrial production: the programmable

logic controller in 1968 and the multifunction robot in 1959. The first computers and the Internet also appeared in the 1990s. Manufacturing operations became increasingly complex and automation gained in strength in order to relieve humans of the most dangerous tasks. This revolution has enabled organisations to evolve in the search for performance, productivity, cost, deadlines and quality with continuous improvement.

In the period 1900-1960, it was the era of production, and quality made its appearance in Japan and then in the United States. Maintenance appears more and more and leaves out servicing. Driving and mining machines were lubricated and maintained. For certain sensitive systems, such as the first boilers, new regulations were introduced with controls, interventions and the first steps towards preventive maintenance. The 1950s saw the advent of reliability-based maintenance in the nuclear industry, while Total Productive Maintenance (TPM) by Toyota, was introduced in the 1970s.

In the 1980s and 2000s, maintenance made many advances, such as the appearance of reliability-based maintenance and the arrival of the first Computerized Maintenance Management System (CMMS) and Enterprise Resource Planning (ERP) computer systems. These made it possible to process larger volumes of data, to automate decision-making procedures and to keep up with the constant evolution of performance criteria linked to competitions and market demands [Julien and Martin, 2021] and [Tahiri, 2021].

1.3.4 Industry 4.0

This industrial revolution is only just beginning to emerge, and represents the factory of the future as a global, inter-communicating system with flexible, integrated and connected factories. All the building blocks to achieve this already exist with: IOTs, automatons, robotics, artificial intelligence, big data, the cloud, renewable energies, energy storage etc. This revolution is essentially based on two concepts: the digital factory and digital manufacturing. The digital factory integrates the dematerialisation and interconnection of products, machines and people with instantaneous communications that allow flexible and innovative production, agile, intelligent with personalised services... Digital manufacturing integrates mechatronic technologies, artificial intelligence and creative techniques such as cobotics, 3D printing...

Maintenance today still uses methods that were born with the industrial revolutions

such as corrective and preventive maintenance. The rise of new technologies with even more interconnections, intelligence, and data flow exchanges make it possible to establish predictive maintenance in order to predict when there will eventually be a failure and the maintenance intervention strategy to be put in place [Julien and Martin, 2021] and [Tahiri, 2021].

1.3.5 Maintenance objectives

The vocation of the maintenance is to ensure in all the sectors of activity the following objectives [Yves, 1992] [Tahiri, 2021] in Figure 1.2: The main objective for the

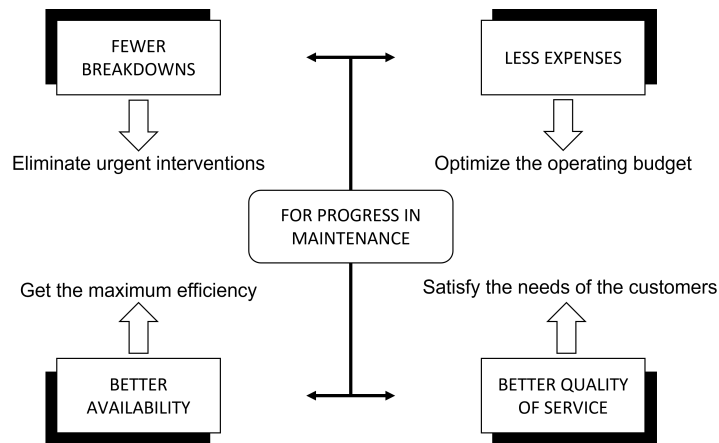


Figure 1.2 – The four objectives of maintenance [Yves, 1992]

progression of maintenance is to ensure that equipment and systems operate reliably and efficiently to ensure customer satisfaction and company profitability. This involves several objectives, such as reducing the number of breakdowns by eliminating urgent maintenance interventions through repairs and implementing preventive maintenance. By reducing the number of breakdowns, the company can also reduce the costs associated with equipment repair and replacement. Maintenance costs represent a significant portion of the total operating costs of any production or manufacturing facility. It also depends on the area of operation, but the costs can represent between 15 and 60% of the overall cost of goods produced. On average, maintenance costs for the food industry are 15% of the goods produced, while for heavy industries such as iron, steel and paper, they represent 60% of the total cost of production [Mobley, 2002]. Indeed, the maintenance costs can be divided into two major classes: direct and indirect maintenance costs. The direct maintenance costs are the costs paid directly for maintenance actions, such as spare part cost, labor

cost, setup scaffolding, buying/renting tools for conducting maintenance, etc. The indirect maintenance costs are the costs associated with stopping the system for maintenance, such as production loss cost, downtime cost, lost sales due to delays/quality issues, rework and defects cost [Thomas, 2018]. Maintenance also helps to improve service quality and equipment availability, ensuring that customers receive high quality service, that their needs are met, and that equipment is available when needed for maximum performance.

1.3.6 Different types of maintenance

In this section, we depict the various types of maintenance as represented in Figure 1.3, which can be broadly categorized into corrective, preventive, and improvement maintenance.

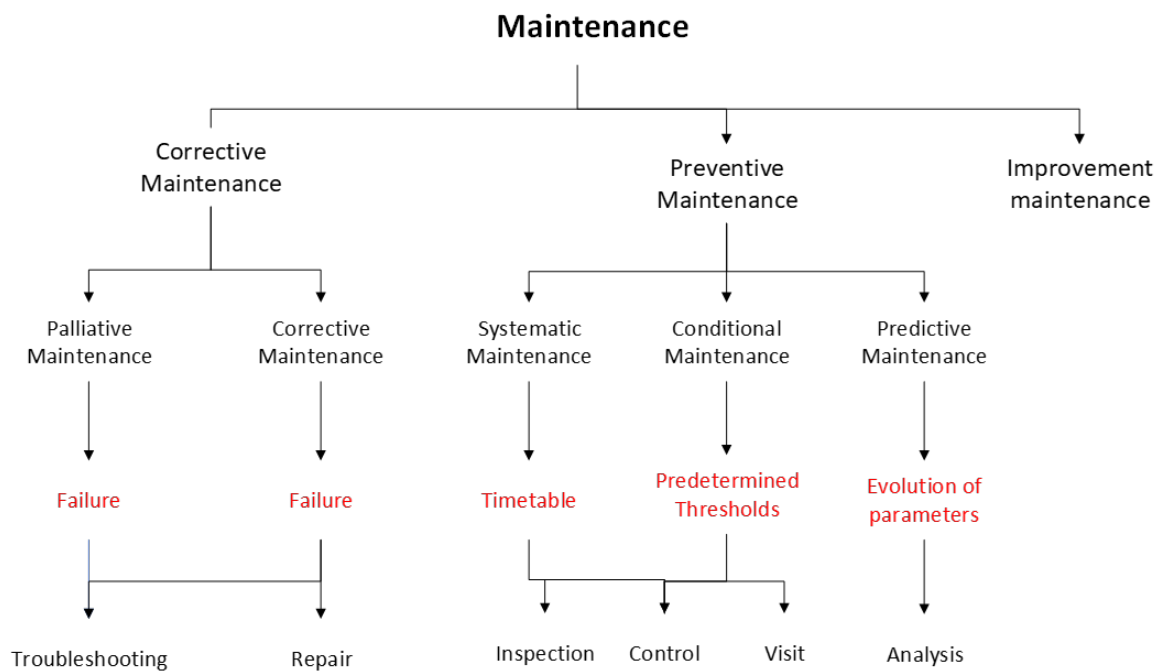


Figure 1.3 – Maintenance Types

1.3.6.1 Corrective maintenance

Corrective maintenance corresponds to the operation undertaken following a failure. Its objective is to get the system back into operation by troubleshooting, repairing or replacing defective parts. There are two types of corrective maintenance:

- Palliative maintenance, which corresponds to troubleshooting with the aim of temporarily restoring a system to operation, so that it will function in degraded mode until it is repaired. It thus makes it possible to avoid a total stop in production.
- Curative maintenance is different, it directly treats the causes and consequences of the breakdown, and allows a system to be repaired in the long term by replacing the defective parts with new ones. The system returns to service under normal conditions.

1.3.6.2 Preventive maintenance

In the definition of preventive maintenance according to [Heng, 2017], we include all controls, visits, maintenance interventions carried out preventively. Preventive maintenance is opposed to corrective maintenance triggered by disturbances or events, and therefore suffered by maintenance. It includes :

- Systematic controls or visits.
- Expertise, actions and replacements carried out following controls or visits.
- Systematic replacements.
- Conditional maintenance or non-destructive testing.

The preventive maintenance consists in following the evolution of the state of a system, component, so as to foresee an intervention without a reasonable delay and the purchase of necessary part.

Systematic maintenance is carried out according to a schedule depending on the time or the number of uses. At each visit, the state of the system is determined either by a measurement value (thickness, temperature, intensity, etc.) or by a visual appreciation. It is possible to interpret the evolution of the state of a system by degrees of appreciation: nothing to report, beginning of degradation, advanced degradation and danger. As a matter of principle, systematic preventive maintenance is carried out according to the conditions that reflect the state of evolution of a failure. The intervention can be scheduled just in time before a failure occurs.

Conditional maintenance is, according to the definition of the Afnor, "maintenance subordinated to a type of predetermined event (self-diagnosis, information coming from a sensor, measurement...)". It allows continuous monitoring of the equipment in service

and the decision to intervene is taken when there is experimental evidence of an imminent defect or a predetermined degradation threshold. During the design of an installation, the designer defines tolerances for certain parameters. The progressive variation of these parameters does not imply the failure of a component. But when a parameter goes out of the prescribed tolerance, the operation may be disturbed. The monitoring of the evolution of the parameters allows to specify the nature and the date of the interventions. The monitored parameter can be :

- An electrical measurement (voltage, current...).
- A temperature measurement.
- A percentage of particles in the oil.
- A vibration level

The parameter to be monitored must therefore be chosen to best characterize the degradation of the components or the cause of the operating disturbance.

1.3.6.3 Improvement maintenance

Improvement maintenance is designed to improve the performance and functionality of an existing system, and is of paramount importance to organizations seeking to optimize their use of these systems. This form of maintenance can take many forms, including software upgrades, hardware improvements or even a complete system redesign. Advances in technology can quickly render systems obsolete, making improvement maintenance essential for companies looking to stay competitive. In short, improvement maintenance allows companies to maximize the use of their existing systems while staying current with the latest technologies and best practices.

1.3.6.4 Predictive maintenance

Predictive maintenance in the Internet of Things (IoT) era can be summarized as a maintenance methodology that brings together the power of machine learning and streaming sensor data to maintain machines before they fail, optimize resources, and thereby reduce unplanned downtime. That autodetects the onset of degradation or deterioration prior to failure by assessing the current and future health state of the system, thereby providing an opportunity to investigate and subsequently eliminate causal factors by taking preventive actions. The fundamental difference between a preventive maintenance program and a predictive maintenance program is that the latter is based on assessing

the actual health of the system, instead of relying on statistical averages to determine maintenance intervals. Predictive maintenance is investigated in scenarios where maintenance costs need to be controlled without jeopardizing the availability of the system in question [Shetty, 2018]. Past studies have estimated that a properly functioning predictive maintenance program can provide savings of 8–12% over and above a program that uses preventive maintenance alone [Sullivan et al., 2010].

The potential of predictive maintenance can be demonstrated through the concept of the P-F curve as shown in Figure 1.4. This curve represents the behaviour of the lifetime of a system that deteriorates with time, leading to functional failure. This approach provides a time window of opportunity, i.e. the PF interval, between the moment of potential failure (P), which is the moment when the system may be failing, and functional failure (F), which is the moment when the system actually fails. The key to effective maintenance is to maximise the P-F interval, thereby increasing the time available to make an informed decision and take action.

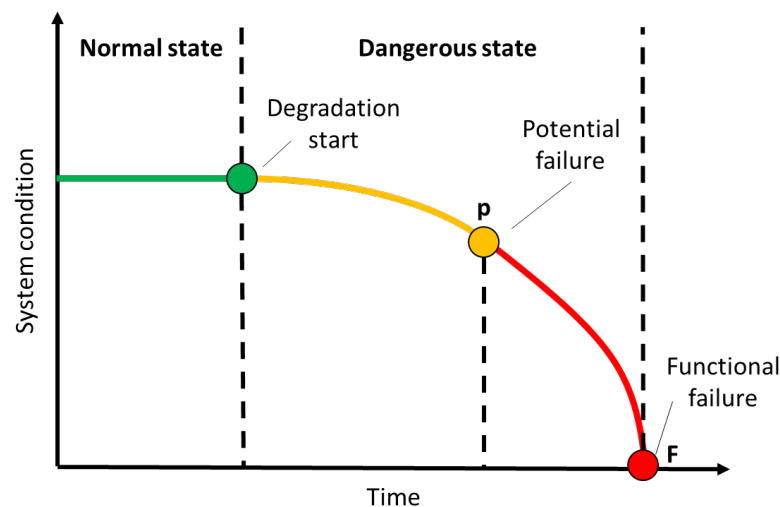


Figure 1.4 – Predictive maintenance in a context of P-F curve adapted from [Bousdekis and Mentzas, 2021]

Corrective maintenance is implemented after the F-point, preventive maintenance allows specific time intervals to be set to avoid the F-point while predictive maintenance can take advantage of big data to always stay as close as possible to the onset of degradation and support cost-effective decisions even before the symptoms of failure are visible to humans [Bousdekis and Mentzas, 2021] and [Shetty, 2018].

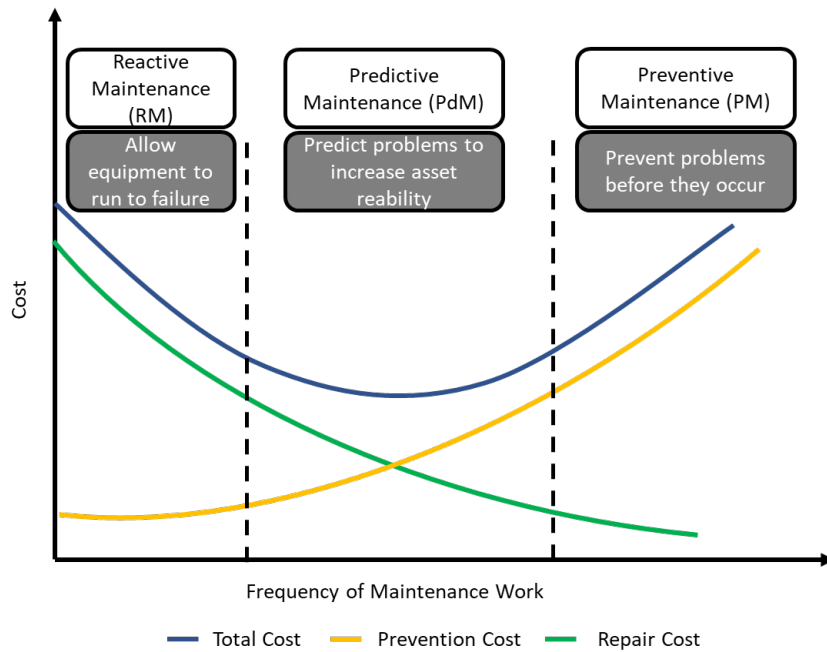


Figure 1.5 – Comparison of RM, PM and PdM on the cost and frequency of maintenance work from [Ran et al., 2019]

Predictive Maintenance, aims to predict when the equipment is likely to fail and decide which maintenance activity should be performed such that a good trade-off between maintenance frequency and cost can be achieved in Figure 1.5 [Ran et al., 2019]. It is observed that reactive maintenance has the lowest cost of prevention due to failure management and preventive maintenance has the lowest cost of repair due to downtime planning. While predictive maintenance allows avoiding unplanned reactive maintenance without incurring the costs associated with preventive maintenance. Finally we have listed the advantages and disadvantages in Table 1.1.

Table 1.1 – Advantages and disadvantages of different maintenance techniques - Adapted from [Abbasi et al., 2018] and [Shetty, 2018]

Types of maintenance	Pros	Cons
Curative maintenance	<ul style="list-style-type: none"> • Minimal staff needed. • Minimal planning required. 	<ul style="list-style-type: none"> • Extremely costly. • Safety risk for employers. • Unpredictable failures
Preventive maintenance	<ul style="list-style-type: none"> • Lower capital cost. • Reduce breakdown. 	<ul style="list-style-type: none"> • Requires planning time. • Repairs even when there is no fault. • Invasive (causing incidental damage). • Labor-intensive.
Predictive maintenance	<ul style="list-style-type: none"> • Increase reability. • Increase life-cycles. • Optimized maintenance activities and cost. • Optimized spare parts. 	<ul style="list-style-type: none"> • Increased investment in diagnostic equipment. • Staff training required for the analysis of data.

1.4 The state of data in maintenance

In this section, we delve into the state of data in maintenance across its various stages, including sourcing, collection, storage, analysis, transfer, and management. We explore the different advancements in maintenance, as represented in Table 1.6, and their impact on data handling in the maintenance process.

1.4.1 Data in Maintenance 1.0

At the time of the first industrial revolution, machines were mechanical, the only objective was that the system should work, there was no method, no specific maintenance tools, nor any form of data collection. The users used the machines until they malfunctioned or broke down completely. At that precise moment, they did their best to repair them, and as they did so, they gained experience and passed on maintenance skills.

	Maintenance 1.0	Maintenance 2.0	Maintenance 3.0	Maintenance 4.0
Data source	Operator experience	Maintainer and machines	Operator, maintainer, machines, information and computer systems	Operator, maintainer, information systems, OEM and suppliers data
Data collection	Manual collection	Manual collection	Semi-automated collection	Automated collection via sensors and IoT
Data storage	Operator memory	Written documents	Databases	Cloud services
Data analysis	Arbitrary	Reliability theory based on Bathtub curve assumption	Conventional algorithms	Fuzzy logic, neural networks, evolutionary algorithms, machine learning
Data transfer	Verbal communication	Written documents	Digital files	Digital files
Data management	N/A	Human operators	Information systems	Cloud and artificial intelligence

Figure 1.6 – Comparison of maintenance data in different maintenance ages, inspired from [Jasiulewicz-Kaczmarek et al., 2020]

1.4.2 Data in Maintenance 2.0

So it was the second industrial revolution that made it possible to have production lines, the interest was to produce in mass and for that, it was necessary to pass to a new mode of maintenance. The machines were equipped with analogue sensors, the operator had to take readings and enter them by hand in maintenance documents. The first theories on reliability analysis and bathtub curve on the maturity of a system were born.

1.4.3 Data in Maintenance 3.0

The 3rd industrial revolution with the birth of electronics and computers has led to a huge advance in the field of digital data. Everything is computerised, production data, sensors, actuators, processes are automatically recorded in local databases. The maintenance operator only enters the information for his intervention in a CMMS system on local servers.

1.4.4 Data in Maintenance 4.0

With the birth of the Internet, the miniaturisation of electronic components and intelligent sensors/actuators (Industrial Internet of Things - IIOT), maintenance is more

efficient and more complex than ever. Data can come from the operator, maintenance, the manufacturer, the production system... It is collected automatically and stored in local databases or in the cloud. It is now possible to analyse it with key performance indicators, to run processing or machine learning algorithms. This makes it possible to take decisions for maintenance actions or to allow the system to be autonomous.

1.5 The effectiveness of data-driven predictive maintenance

Data-driven predictive maintenance is a technique used in industrial maintenance to predict when equipment may fail through data analysis. Machine learning algorithms are used to analyse large amounts of data from sensors, actuators or other sources to identify patterns and anomalies that can be used to predict when equipment failure is likely to occur.

In the study conducted [Ngu et al., 2019] shows us the effectiveness of predictive maintenance compared to simple maintenance. To be efficient and competitive, industrial installations must no longer consider that equipment failures occur after a specific service life. For instrumentation and process control, it is crucial to have sensors, actuators and end systems that operate within the operating limits defined by the manufacturer to ensure the safety and availability of the manufacturing process. This article also shows that instrumentation can provide data for predictive maintenance analysis on equipment such as: gas turbines, steam turbines, motors, compressors, heat exchangers and process conditions that are key elements and must always be available and reliable. With all these conditions, it is possible to deploy predictive maintenance on the studied system.

[Kanawaday and Sane, 2017] use a machine learning method called AutoRegressive Integrated Moving Average (ARIMA) to predict bad production runs on a slitting machine. The system configuration consists of a Programmable Logic Controller (PLC) for storing data of parameters to be monitored such as time stamp, pressure, tension, diameter and width. This data is transmitted to an Industrial Personal Computer (IPC) which then sends it to the cloud using the Message Queuing Telemetry Transport (MQTT) protocol. By analysing the data, they find that there is a quality defect when there is a pressure drop, which leads to a bad production cycle. The prediction results show that the artificial neural network is more efficient with an accuracy of 98.69% compared to Naive Bayes 96.61%, SVM 95.52% and CART 94.46%.

We have taken some bibliographical references from the review from [Zhang et al., 2019b] of machine learning techniques used to perform data-driven predictive maintenance on real-life cases which we have detailed and then summarised in Table 1.2

1.5.1 Logistic regression

Firstly, we will present the work dealing with predictive maintenance with the application of logistic regression, where we present the concept in the tools and methods section 2.2.3.2. [Li et al., 2015] proposed a method using logistic regression (LR) combined with acoustic emissions and the cutting force signal which are used for the control of the cutting process. Two models are competed to evaluate the reliability: model 1 consists of the cutting force, the normalized energy of the wavelet packet of frequency bands 2 and 7, while model 2 does not contain the cutting force. Model 1 obtains the best evaluation, but it is difficult to obtain the cutting force in practice so model 2 is more practical.

[Yan and Lee, 2004] present a method for monitoring the condition of lift doors using the logistic regression model on 220 continuous door opening and closing cycles for model validation. They used the wavelet packet energies as features and Fisher's criterion to select the features and finally the maximum likelihood estimate as parameters of their model. They defined a confidence value noted CV, if the value is close to 1 then the operation is normal, if it is close to 0 then the system is about to fail. The threshold for classification as good or bad is set at 0.8. In all three models tested, they introduced faulty cycles that were successfully detected.

A datadriven modeling method was demonstrated in [Nor et al., 2021] for the health condition monitoring of gas circulator units by using a combination of LR and L2-Support Vector Machine (L2-SVM) models.

[Pandya et al., 2014] focused on fault diagnosis of rolling element bearings. The features for classification are extracted through wavelet packet decomposition using RBIO 5.5 wavelet. The whole classification is done using two features: energy and kurtosis. Four bearing conditions such as bearing with outer race defect, inner race defect, ball defect and combined defect on outer race, inner race and ball have been used in this paper. Out of the three classifiers considered, namely ANN, SVM and MLR, the latter was found to be the most effective with a classification accuracy of 100% and a short computational time of only 0.33 seconds.

This study, by [Caesarendra et al., 2010], proposes the combination between relevance vector machine (RVM) and LR in order to assess the failure degradation and prediction

from incipient failure until final failure occurred. High kernel-width gave serious overfit prediction of failure probability or RVM testing. The best RVM testing performance was reached at kernel-width 0.0001 with RMSE and R being 0.0002 and 0.99, respectively. In addition, selection of kernel-width with value less than 0.0001 affected high error and low correlation of failure probability prediction. Finally, the prediction of failure time allows a prediction accuracy of 99.49%.

[Zhang et al., 2018] worked on a vibration-based monitoring method for online detection of punch failure in the micropiercing process. The signal features are extracted from both time and frequency domain signal. The statistical overlap factor is utilized to select the best features. To evaluate the tool condition, a LR model is used with the selected features as an input. Among the developed models, the logistic models by using the selected features separately give better accuracy, over 99%.

The approach proposed by [Skordilis and Moghaddass, 2017] uses Kalman filter for degradation state estimation and LR for failure/hazard progression tracking.

The study conducted by [Ahmad et al., 2019], a dynamic regression models based approach was proposed for the RUL estimation of bearings. These models were used to capture the evolving trend in the bearing's health indicator. The performance of the proposed method was compared to other schemes for RUL estimation and time start prediction detection using the same run-to-failure test data. The proposed approach achieved reasonably good results in comparison to existing methods.

[Phillips et al., 2015] claim that LR models are likely to be an accessible concept for working engineers, unlike ANN and SVM which often require specialist training. The performance of LR, cascade-correlation neural network (CCNN) and SVM models has been evaluated for classification of oil sample data from mining trucks. Identical data sets and explanatory variables are used in all three models. The LR model demonstrated an ability to classify approximately 89% of the oil samples correctly and LR outperformed both the CCNN and SVM models in terms of predictive performance.

In the paper proposed by [Yu, 2018], logistic probability generated by LR with manifold regularization is used to serve as a comprehensible indication to assess tool health state. Prognostic features are selected firstly by LR with penalization regularization (LRPR) to improve the performance of the proposed tool health prognostics system. The experimental results based on the full life datasets from a tool test-bed indicate that the proposed scheme is effective for tool health prognostics.

1.5.2 Support Vector Machine

Now we will present the work dealing with predictive maintenance with the application of SVM, where we present the concept in the tools and methods section 2.2.3.2. The approach of [Widodo et al., 2009] on fault diagnosis of low-speed bearing using multi-class RVM and SVM. The data was acquired from the low-speed bearing test rig using acoustic emission and accelerometer sensors under a constant load with different speeds. Component analysis was performed initially to extract the features and to reduce the dimensionality of original data features. It has also been demonstrated that SVM outperformed an RVM for the diagnosis of low-speed bearings.

[Susto et al., 2015] propose a multiple classifier predictive maintenance methodology for replacing tungsten filaments used in ion implantation, which is one of the most important processes in semiconductor manufacturing fabrication. According to the different comparison results, the multiple classifier predictive maintenance-SVM obtain better performance than multiple classifier predictive maintenance-KNN and predictive maintenance radial basis function in terms of unexpected breaks and maintenance costs.

[Baptista et al., 2018] propose a framework that can predict when a component/system will be at risk of failure in the future, and therefore, advise when maintenance actions should be taken for airline maintenance scheduling purpose. They use an autoregressive moving average model (ARMA) along with data-driven techniques. The overall comparison of the methods shows SVM obtained the best results.

An approach is presented by [Shafi et al., 2018] for fault prediction of four main subsystems of vehicle, fuel system, ignition system, exhaust system, and cooling system. All data that we have collected comes from 70 vehicles of Toyota Corolla type. Four classifiers were compared with decision tree, random forest, KNN and support vector machine. SVM obtain the best accuracy 96.6% (ignition system), 98.5% (fuel system), 98% (exhaust system), and 96.6% (cooling system).

[Zhang et al., 2015] present a novel hybrid SVM model for fault detection and classification of motor bearing. This hybrid model optimized by inter-cluster distance in the feature space is used to classify the fault type as well as fault severity. The results demonstrate its effectiveness and robustness for motor bearing fault detection and classification.

[Nieto et al., 2015] propose a SVM hybrid model that accurately predicts the RUL of the gas turbines. Furthermore, the model performed requires no knowledge of the previous status of the aircraft engine, as it only requires information of the current situation of the same data. The presented research uses data corresponding to an aircraft engine of 90,000

lb thrust class. The data includes working conditions with heights from sea conditions to 40,000 ft and temperatures from 51°C to 39°C.

1.5.3 Decision Tree and Random Forest

Now we will present the work dealing with predictive maintenance with the application of Decision Tree (DT) and Random Forest (RF), where we present the concept in the tools and methods section 2.2.3.2. [Li et al., 2018a] claim an improved DT-based fault diagnosis method for practical variable refrigerant flow system. This model was combining DT model with virtual sensor-based fault indicators. Results indicate that the method correctly isolates the three faults: airside fouling, refrigerant undercharge and overcharge. The improved DT method is also compared with three tree-based data-driven methods including CART, RF and generalized boosted regression. Its overall fault diagnosis accuracies are 94.44% and 80.55% for the experimental and the on-line testing datasets, respectively.

The work of [Canizo et al., 2017] consist of a predictive model generator for each monitored wind turbine by means of RF algorithm. In addition to this, a monitoring agent that makes predictions every 10 minutes about failures in wind turbines during the next hour. Finally, they implemented a dashboard where given predictions can be visualized. Their approach allows for a better accuracy of 82.04% compared to [Kusiak and Verma, 2011].

[Li et al., 2016] address the use of a deep random forest fusion technique to improve fault diagnosis performance for gearboxes by using measurements of an acoustic emission sensor and an accelerometer that are used for monitoring the gearbox condition simultaneously. The proposed DRFF technique is evaluated using gearbox fault diagnosis experiments under different operational conditions, and achieves 97.68% of the classification rate for 11 different condition patterns. Compared to other peer algorithms, the addressed method exhibits the best performance.

1.5.4 Artificial Neural Network

Now we will present the work dealing with predictive maintenance with the application of ANN, where we present the concept in the tools and methods section 2.2.4.1. [You et al., 2014] proposed an innovative approach to realize laser welding process monitoring and welded defect diagnosis. First, the proposed method proves that the high-cost sensor with

complicated structure can be replaced by the low-cost and simply structured one after the data-driven model has been established by applying multivariate statistics and machine learning methods. A correlation model concerning optical features and geometrical parameters was established based on Feedforward Neural Network (FNN). Experimental results show that, by using a low-cost sensor with simple structure, as well as integrating FNN and SVM, which are based on Wavelet Packet Decomposition-Principal Component Analysis (WPD-PCA), effective estimation and classification on welding status and defects can be realized.

[Prieto et al., 2012] has introduced a novel diagnosis methodology applied to bearing faults using information in time domain from the vibration data. Due to the different number of considered operating conditions (25) and the number of considered bearing fault scenarios (six), a two-level hierarchical neural network is applied to assure the optimum pattern recognition. Specifically, a hierarchy of multilayer perceptrons (MLPs) (hMLP) has been developed and the classification ratio for the test set is 95%.

[Ahmed et al., 2015] has proposed an engine fault detection and classification technique using vibration data in the crank angle domain. These data are used in conjunction with Artificial Neural Networks (ANNs), which are applied to detect faults in a four-stroke gasoline engine built for experimentation. A comparative study is provided between the popular backpropagation (BP) method, the Levenberg–Marquardt (LM) method, the quasiNewton (QN) method, the extended Kalman filter (EKF), and the smooth variable structure filter (SVSF). A fully connected feedforward MLP is used with a number of input neurons representing sampled vibration data in the crank angle domain, two hidden layers with eight neurons each, and eight output units representing the network classification results. For the SVSF-based method, the network achieved 100% training accuracy and 97% testing accuracy.

1.5.5 Deep Neural Network

[Hu et al., 2017] adopted the deep neural network to recognize faults in bogies. A bogie is a carriage located under a railway vehicle, on which the axles (and thus the wheels) are fixed. Each health condition includes ten kinds of different speeds. There are eight kinds of different health conditions in total. Each sample includes 16 time-domain and frequency-domain features. The input layer of each fault diagnosis will have 1280 (10 8 16) nodes. The model is made up of five layers for the network, and neurons of each layer are 1280, 600, 300, and 8, respectively. The diagnostic accuracy rate of deep neural

networks when the high-speed train is under normal conditions can reach 100%.

The work of [Wang et al., 2017] consist of a deep neural network (DNN)-based framework being developed to monitor conditions of WT gearboxes and identify their impending failures. Six data-mining algorithms, the k-nearest neighbors, least absolute shrinkage and selection operator, ridge regression (Ridge), support vector machines, shallow neural network, as well as DNN, are applied to model the lubricant pressure. In the comparative analysis of prediction models, the DNN model offered more accurate prediction results.

The behavior and interaction of the main components of Ship Propulsion Systems cannot be easily modeled with a priori physical knowledge, considering the large amount of variables influencing them. [Cipollini et al., 2018] investigate the problem of performing condition-based maintenance through the use of data-driven model and compare their model to others supervised learning techniques. The proposed models are quite effective and reliable.

1.5.6 Auto-Encoder

The proposed method by [Shao et al., 2017] is applied to the fault diagnosis of rotor and electrical locomotive bearing. The results confirm that the proposed method based on new deep Auto-Encoder (AE) is much more effective and robust for feature learning and fault diagnosis than shallow learning methods. Compared with standard CNN, the proposed method shows slightly better performance.

[Verma et al., 2013] compare the performance of sparse AE with soft max regression, fast classifier based on Mahalanobis distance and SVM in fault diagnosis of air compressors. The performance of sparse autoencoders was found to be better than SVM and Mahalanobis distance based classifiers with accuracy up to 97.22%. Not only was it more accurate than SVM, but also faster with regard to training times.

[Sun et al., 2016] present a Deep Neural Network (DNN) approach for induction motor fault diagnosis. The approach utilizes sparse AE to learn features. In this study only one-layer SAE is utilized to realize the DNN for the induction motor fault diagnosis. In the proposed sparse AE-based DNN framework, the input size, hidden node and output size are set to 2000, 600 and 2000, respectively. Considering that vibration signals represent six different working conditions, the layer sizes for DNN is set as [2000 600 6]. Then, the proposed approach is also compared with two classifiers built on the top of the SAE with an average accuracy of 97.61%.

1.5.7 Summary table

The Table 1.2 summarizes all the previous works sorted by type of algorithm.

Table 1.2 – Summary of the data-driven predictive maintenance method using three types of machine learning algorithms from [Zhang et al., 2019b]

Algorithm	Reference	Signal	Application Scenario	Task	Accuracy	Datasets
LR	[Li et al., 2015]	Acoustic emission	Cutting tools	Reability evaluation	99.97% and 99.96%	Dongyu Machine and Tool CMV-850A Center
	[Yan and Lee, 2004]	vibration / current	elevator door system	degradation assessment	100%	from the door control board
	[Nor et al., 2021]	vibration	gaz circulation units	health state estimation and RUL prediction		From EDF Energy
	[Pandya et al., 2014]	vibration	bearings (rotating)	fault degradation estimation	100%	from the test rig
	[Caesarendra et al., 2010]	vibration	bearings	estimate failure degradation	99.49%	from simulation by MATLAB and test rig
	[Zhang et al., 2018]	vibration	the micro-piercing process	failure detection	99%	from the actual machine
	[Skordilis and Moghaddass, 2017]			RUL prediction	99%	from the simulated dataset and real industrial settings
	[Ahmad et al., 2019]	vibration	bearings	RUL prediction	69% (average)	from the PRONOSTIA platform
	[Phillips et al., 2015]	oil samples	the diesel engine	fault diagnosis	89%	from actual engines
	[Yu, 2018]	acoustics and vibration	cutting tools	RUL prediction		from actual engines
Support Vector Machine	[Widodo et al., 2009]	acoustic and vibration	bearings	fault diagnosis	97.96%	from the test rig
	[Susto et al., 2015]	pressure, voltage, current	tungsten filaments	fault diagnosis	98.52%	from a implanter tool
	[Baptista et al., 2018]		valves of the aircraft	fault prediction		from the real engine bleed valve
	[Shafi et al., 2018]	electric	four main sub-systems of vehicle	fault prediction	96.6%, 98.7%, 98%, 96.6%	from sensors on Corolla cars
	[Zhang et al., 2015]	vibration	bearings	fault detection and classification	97.75%, 97.91%	provided by CWRU
	[Nieto et al., 2015]	temperature, pressure and speed	aircraft engines	RUL prediction		from the actual aircraft engine

1.5. The effectiveness of data-driven predictive maintenance

Algorithm	Reference	Signal	Application Scenario	Task	Accuracy	Datasets
Decision Tree and Random Forest	[Wu et al., 2016]	force, acoustic and vibration	tool wear in dry milling operations	performance evaluation	99.2%	from [Kusiak and Verma, 2011]
	[Canizo et al., 2017]	status data (activations and deactivations)	wind turbines	fault prediction	82.04%	gathered from wind turbines during two years
	[Li et al., 2016]	acoustic and vibration	gearbox	fault diagnosis		from test bed
Artificial Neural Network	[You et al., 2014]	optical and visual	laser welding	process monitoring and fault diagnosis	-	from a high-power (16 kW) disk laser welding system
	[Prieto et al., 2012]	vibration	bearings	fault diagnosis	95%	from an induction motor
	[Ahmed et al., 2015]	vibration	a four-stroke gasoline engine	fault detection and classification	97%	from Ford's Powertrain Engineering Research and Development Centre (PERDC)
	[Hu et al., 2017]	vibration	high speed train system	fault diagnosis	100%	from multi-body dynamic software SIMPACK
	[Wang et al., 2017]	lubrifiant, pressure	wind turbine gearbox	identify impeding failures	-	provided by 6 commercial wind farm in China
	[Cipollini et al., 2018]	speed, thrust and torque	gaz turbine	Condition-based maintenance	-	from University of California in Irvine (UCI)
Auto-Encoder	[Shao et al., 2017]	vibration	rotor and bearings	fault diagnosis	97.10%, 95.23%, 100%	from the rotor test rig, CWRU, NASA
	[Verma et al., 2013]	acoustic	motors	fault diagnosis	97.22%	from an actual air compressor
	[Shafi et al., 2018]	vibration	induction motor	fault classification	97.61%	from a machine fault simulator

1.6 Conclusion

In this chapter we have presented the main concepts of maintenance necessary to understand the context and the problematic that we address in this thesis. We have outlined the definitions of maintenance, its evolution through the industrial revolutions, its objectives and the different types that exist. We then discussed data-driven predictive maintenance and summarised a body of work in the literature to provide a summary table to show its effectiveness.

Predictive maintenance has an component called time series anomaly detection which we have represented in the next chapter to identify early warning signs of failures.

ANOMALY DETECTION IN TIMES SERIES

2.1 Introduction

Predictive maintenance is a method of predicting system failures using real-time data on system performance and usage. In this chapter, we essentially focus on two main questions: how can we do predictive maintenance? To answer this question, we will study together the tools and methods used. Followed by a second question: what kind of predictive maintenance can we implement? We shall first answer the question by the part concerning the detection of anomalies followed by the analysis of the remaining useful life time of a system treated in the literature.

2.2 Tools and methods

The techniques used in predictive maintenance for the two cases of anomaly detection and remaining useful time are equally based on data science. Data science is a science composed of a set of tools and methods that enable to extract from a mass of data, also called big data, significant data, for different areas. This science has become vital because of the rise of new technologies produces exorbitant quantities of raw data and unexploitable in their raw state. This science provides access to a set of mathematical tools, programming and machine learning methods to meet the context and the need of the study. We proposed here a non-exhaustive list of tools:

Programming tools :

According to [Huddar and Kulkarni, 2018] and [Larose and Larose, 2019] Python and R are well known as programming tools for data science. The article by [Bloice and Holzinger, 2016] provides a fundamental understanding of the tools Python offers for Data Science and Machine Learning tasks. It presents different packages for tasks such as data preparation, exploratory analysis, modeling and evaluation.

- Python is an open source programming language used in many fields such as software, data science, web development and video games. It was born in 1991 and has become one of the most popular programming languages in the world so much so that it holds the first place in several popularity indexes like TIOBE [Tiobe Index, 2021] and PYPL [PyPL Index, 2021]. It is supported by a large community of developers and users who ensure its growth, optimization and the publication of new libraries for all types of uses. This programming language is easy to learn because it is easy to read and write because it has many similarities with human language. This concept is the core of its design, it is even highly recommended to start with this programming language if you want to do programming for the first time. In the last few years, Python has increased its popularity in the field of data science thanks to the points mentioned above but also because of its gigantic library and package proposal to implement of all the phases of a data science project such as data analysis, processing, visualization, machine learning.
- R is an open source programming language created for statistical calculation and graph generation. It was created in 1992 and is widely used in scientific and academic research. R is as popular as it occupies the 11th and 7th place in the TIOBE [Tiobe Index, 2021] and PYPL index [PyPL Index, 2021]. It is possible to use R in complex functions in only a few lines of code and it offers many tests, statistical models easily available and usable as clustering, classification, linear and nonlinear modeling. A large part of the tools offered by R is due to its community, which makes it possible to have at disposal one of the richest libraries of data science packages available through the Comprehensive R Archive Network (CRAN). What makes it popular is also the fact that it can generate quality reports by supporting data visualization and data analysis.

Comparison between Python and R

The following comparison, based-on [Ozgur et al., 2017] [Johnson and Chandran, 2021] [Rahmany et al., 2020], is made with the objective of showing the strengths and weaknesses of each of these two programming languages in data science. Although both are adapted for data science, Python is more used by the community than R because it is a language whose objectives are more conducive to other application domains while R is specific to statistical analysis.

Table 2.1 – A comparison between Python and R

Criteria	Python	R
Release date	1991	1992
Open source	Yes	Yes
Ecosystem	Over 300 000 packages available on PyPI	Approximately 19 000 packages available at CRAN
Popularity	Python is more popular because it has a large community of developers and programmers	R is less popular among these users which is composed of scientists, research and development team on data analysis.
Learning level	Easier to learn at first	Requires more time to learn
IDE	Jupyter Notebook, Spyder, Pycharm...	RStudio, StatET...
Advantages & Disadvantages	Easy to learn and quick to set up	More difficult to learn because its syntax is particular without programming experience
	Language for various purposes	Statistical language focused on analysis and visual representation of data
	Python's statistics packages are less powerful	The statistical packages are very powerful
	Representation of lower quality graphics	Visual representation of a quality that surpasses
	Slower but not by much	Faster
	Python is good at building from scratch	R makes it easy to perform mathematical calculations and statistical tests
	Require more maintenance	More robust and easy to maintain

2.2.1 Programming libraries for data science

As we said before Python and R are adapted to data science and machine learning tasks. According to [Bernard, 2016] and [Wickham and Grolemund, 2016] we therefore have at our disposal several essential libraries to deal with them such as :

Python

- Pandas : perfect for data handling and data processing tools
- NumPy : offers a collection of scientific calculation functions
- Matplotlib and seaborn : essential, it allows data visualization

- Scikit-learn : provides a large panel of machine learning algorithms
- Keras and Tensorflow : mainly used for deep learning

R

- Dplyr, tidyr and data.table for data manipulation.
- Ggplot2 for data visualization.
- Mlr3 for machine learning.
- Caret for machine learning in classification and regression

2.2.2 Cloud computing

Cloud computing is the practice of using remote computer servers hosted on the Internet to store, manage and process data instead of a local server or a personal computer. This technology is becoming more and more important for those who wish to implement data science and machine learning as data is becoming more and more present, learning models are becoming more complex. This is why a simple user is quickly limited by the capacities of his computer for a lack of RAM or ROM, what about an industrial company? To get around this problem it is possible to buy or assemble one's own calculation server, data storage for machine learning and deep learning, but this may be very expensive in terms of price, calculation resources, and that you have to maintain yourself, ensure backups, updates, security.

To avoid all these problems, Cloud Services is the solution that allows individuals and businesses to access cloud services at affordable prices according to their needs. It is a trend that has spread rapidly in the IT industry that offers the opportunity to store data on remote servers connected to the Internet. To process them with analytical tools for artificial intelligence (AI) and business intelligence (BI). With this solution, the user does not need to buy server hardware, upgrade it and maintain it, but rents it for as long as he wants according to his needs in computing power and memory storage. There are three main cloud computing services [Mell et al., 2011]:

- **Infrastructure-as-a-Service (IaaS)** provides the consumer with processing, storage, networking and other fundamental computing resources where the user can deploy and run software as desired. He does not manage the underlying infrastructure but has control over applications, operating systems, storage and the ability to have limited control over some network components, such as host firewalls. This solution allows him to avoid any hardware purchase.

- **Platform-as-a-Service (PaaS)** allows to make available to the user a software development and deployment environment. Thus, he only has to develop, run and manage his own applications without having to worry about the associated infrastructure.
- **Software-as-a-Service (SaaS)** is the most widespread service. It allows to provide the user with applications that can be manipulated from a web browser. Thus, the user no longer has to manage the availability, security and updates of the service.

New services have also emerged in recent years such as Notebook-as-a-Service (NaaS) and Machine-Learning-as-a-Service (MLaaS), which are mainly used for the implementation of data science projects as well as the implementation of artificial intelligence (AI) solutions at lower cost. The user can easily use machine learning based solutions in various fields such as risk estimation, strategy, logistics optimization... The currently providers are, according to [Osypanka and Nawrocki, 2022], Amazon Web Services, Microsoft Azure, Google Cloud Platform and OVHCloud and many others offer cloud computing solutions.

2.2.3 Machine Learning Methods

There are mainly four methods of machine learning [Burkov, 2019]:

- Supervised learning, the algorithm used is trained on a set of data called labeled. In other words, each input data is associated with a known data, an output. Through its experience, the algorithm can associate new labels for new data. Classification and regression are examples of supervised learning.
- In unsupervised learning, the algorithm is trained on an unlabeled dataset to identify sub-adjacent structures, or hidden patterns in the data. Examples are dimension reduction and clustering.
- Semi-supervised learning is a technique that combines both sparsely labeled and unlabeled data in large numbers. The objective is to use the knowledge contained in the labeled data to improve predictions on the unlabeled data.
- In reinforcement learning, the algorithm learns through trial and error by dynamically interacting with its environment. The goal is to learn an optimal policy to maximize a long-term reward. Autonomous robots, recommendation systems are examples of reinforcement learning.

2.2.3.1 Regression

Regression is widely used to describe and analyze the correlation between data. It is used to make predictions by using the relationships between different dependent and independent variables to design a predictive model. There are basically three types of regression:

Simple linear regression is used to model the relationship between an independent variable and a dependent variable, in order to predict a target variable with an explanatory variable. The general formula is :

$$y = b_0 + b_1 \times x \quad (2.1)$$

With y the target value, x the independent variable, b_0 the intercept coefficient and b_1 the regression coefficient which is in other words the slope of the regression line. The objective is to find the optimal b_0 and b_1 coefficients to reduce the calculated error between the predicted and actual values.

Multiplied linear regression is based on the same principle as simple linear regression, but this method models the linear relationship between several independent variables and the dependent variable. Its general formula is :

$$y = b_0 + b_1 \times x_1 + b_2 \times x_2 + \dots + b_n \times x_n \quad (2.2)$$

Where y is the dependent variable, x_1 through x_n are the n independent variables, and b_0 the interception. b_1 through b_n are the regression coefficients which indicate the variation of y for a unit variation of each of the independent variables, that are determined by the algorithm. The coefficients of the regression are estimated using a least squares method, which minimizes the sum of the squared errors between the predicted and observed values of the dependent variable.

Polynomial regression is a non-linear regression that allows us to establish relationships between variables using polynomials of degree greater than 1. Thus, the general formula is in the form :

$$y = b_0 + b_1 \times x + b_2 \times x^2 + \dots + b_n \times x^n \quad (2.3)$$

where y is the dependent variable, x is the independent variable, b_0 , b_1 , b_2 , ..., b_n are

the regression coefficients and n is the degree of the polynomial. The objective is still the same on the calculation of the prediction error, but now there is also the coefficient of the polynomial and if it is too high, then the model will be overlearning, so it is vital to find a compromise between the predictive quality and the degree of the polynomial.

2.2.3.2 Classification

The machine learning classification method is used to predict which class the data belong to based on their input features. In the field of predictive maintenance, this method can be used to predict whether a piece of equipment is likely to fail or not, based on historical data such as performance, alarms, environmental... There are several algorithms such as :

- **Logistic regression** is a regression method that allows binary classification. It allows to predict the membership of an observation to one of the two classes of the problem.

$$p(y = 1|x) = \frac{1}{1 + \exp(-z)} \quad (2.4)$$

where $p(y=1|x)$ is the probability that y is equal to 1 as a function of the variable x , \exp is the exponential function and z is a weighted combination of the variables x :

$$z = b_0 + b_1 \times x_1 + b_2 \times x_2 + \dots + b_n \times x_n \quad (2.5)$$

where $b_0, b_1, b_2, \dots, b_n$ are the regression coefficients.

- **Decision tree** learning is a supervised learning approach that can be used in data mining, statistics and of course machine learning. A decision tree consists of a set of nodes, branches and leaves. A node represents the characteristics of a data set, the branches represent the decision rules and each leaf represents the final result. An effective decision tree is one where the concentration of a certain group of data is particularly high in the leaves. Decision trees are easy to understand, easy to visualize, easy to interpret, and provide understandable decision rules. A random forest is also a supervised learning algorithm that can be used for both regression and classification. Random forests create decision trees on randomly selected data samples and obtain a prediction for each tree and select the best solution by voting.
- **The support vector machine (SVM)** or wide margin separator, is a supervised learning algorithm. These SVMs were developed in the 1990's [Boser et al., 1992]

whose principle is to separate the data into classes using a maximum margin. The margin is the distance between the separation frontier and the closest samples. This notion of frontier assumes that the data are linearly separable, which is not often the case. To remedy this, there is a trick called kernel trick, which allows to redistribute the points of the description space E to a new description space (redescription) E' , which is generally of higher dimension, using a necessarily non-linear function ϕ where it is possible to perform a linear separation.

2.2.3.3 Statistical techniques

The GESD (Generalized Extreme Studentized Deviate) method [Rosner, 1983] is a statistical approach used to detect one or more anomalies (outliers) in a univariate data set that follows an approximately normal distribution. This method overcomes the main problem of the Grubbs test [Grubbs, 1950] and the Tietjen-Moore test [Tietjen and Moore, 1972], which is that the presumed number of anomalies, named k , must be precise because if it is not, then the results may be biased.

Principal component analysis (PCA) is a technique that allows to reduce the dimensionality of a multidimensional data set. It allows to find a more concise representation of the data by using a reduced number of variables called "principal components".

2.2.3.4 Clustering techniques

The clustering technique is a machine learning method that consists in grouping data by similarity or distance in the form of a cluster. It is an unsupervised learning method whose goal is to extract properties from large quantities of structured or unstructured data, in order to give them a meaning. Structured data refers to data organized in a tabular format with specific fields and columns that are often found in the form of a database that is easy to analyze, manipulate and search. Whereas unstructured data does not follow a predefined format, so it is more difficult to analyze, manipulate. In general, a cluster refers to a collection of data gathered in a group with similarities. The user defines a number k , which is the number of centroids (center of the cluster) and the algorithm assigns each data to the closest cluster while keeping the centroids as small as possible. One of these limitations is that the user has to specify the number of clusters k from the beginning, while other clustering algorithms such as X-means or G-means will best estimate the value of k according to the data but also decide whether the centroid should

be divided by two to optimize the structure, according to the Bayesian Information Criterion (BIC) and Gaussian Test criteria respectively [El Malki et al., 2020].

2.2.3.5 Expert systems

Knowledge-based techniques are also called expert systems. An expert system is defined as "A computer program designed to model the problem-solving ability of a human expert" [Durkin, 1994]. There are mainly two components in an expert system: a knowledge base and an inference engine. The knowledge base contains the high-level knowledge of a human expert on the problem domain. It includes facts, rules, concepts and relations. As for the inference engine, it is the knowledge processor implemented to work with the knowledge available on the given domain. Thus the knowledge base coupled with the inference engine, make it possible to draw conclusions or recommendations [Durkin, 1994] [Prayote, 2007].

2.2.3.6 Pattern extraction

Pattern extraction is a method of identifying recurring structures or hidden trends in the processed data. It thus allows to extract patterns of normal system behavior. A similar method is the extraction of rare patterns among normal behaviors that can be interpreted as anomalies, failures...

2.2.3.7 Spectral analysis

Spectral analysis consists in transforming temporal data into a frequency representation called spectrum. Thus, it is possible to use it for predictive maintenance by comparing the spectra of normal and abnormal data. This method can be used with vibration sensor data.

2.2.4 Deep Learning Methods

2.2.4.1 Artificial Neural Network (ANN)

The artificial neural network is a mathematical model inspired by the structure and behavior of the biological system of the human brain [Schuman et al., 2017], able to solve machine learning problems. The artificial neuron is composed of several inputs that will receive various information each linked to a weight that can be adjusted. These inputs

correspond to the dendrites of a biological neuron, with associated weights, which coincide with the inhibitory or excitatory actions of the synapses in order to amplify or reduce the input signal. At the level of the body (soma), the neuron processes the information by summing it up and if the result obtained is higher than an excitation, or activation, threshold then it sends a discharge called an action potential along its axon towards other biological neurons. The first artificial neuron model was introduced by [Rosenblatt, 1958] called perceptron.

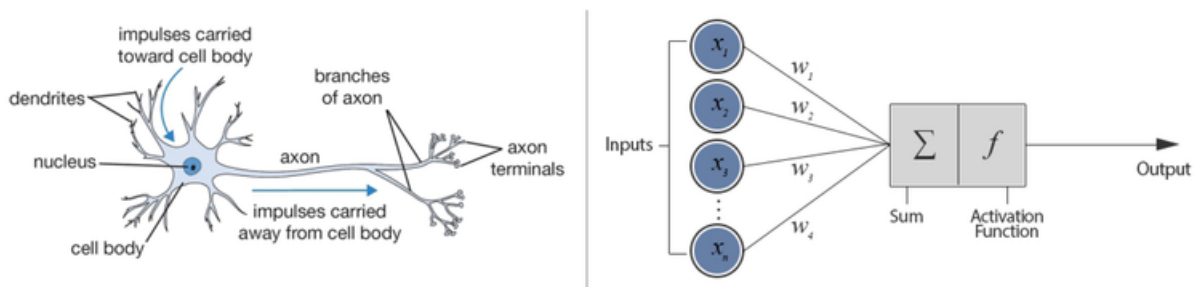


Figure 2.1 – Comparison between biological and artificial neuron from [Mwandau and Nyanchama, 2018]

Perceptron

The perceptron is an artificial neuron (Figure 2.1 that belongs to the family of supervised learning algorithms of binary classifiers, which is used to separate into two classes. It consists of several elements: n inputs (x_1, \dots, x_n) and an output y which is boolean. Each input variable is associated with a weight W , which is a weighting parameter, (W_1, \dots, W_n), to which there is also a bias named b . In the first step, all inputs are multiplied by the weights to which they are attached, we obtain :

$$W_i * x_i \tag{2.6}$$

Then each part is added, we get a weighted sum:

$$\sum_{i=0}^n (W_i * x_i) \tag{2.7}$$

Then each part is added, we obtain a weighted sum to which b is added which has the same role as the intercept in a linear equation. Then the activation function σ , which

allows to activate or not the neuron, is applied on the sum to finally obtain :

$$y = \sigma\left(b + \sum_{i=0}^n (W_i * x_i)\right) \quad (2.8)$$

This activation function allows to transform the result into a number between zero and one which is then transmitted to the next neuron. There are several activation functions such as: identity, binary step, logistic, sigmoid, hyperbolic tangent...etc which have different limits but whose operation remains the same.

Multilayers perceptron

The multilayers perceptron (MLP) (Figure 2.2) is a layered feedforward neural network in which the information flows unidirectionally from the input layer to the output layer, passing through the hidden layers [Bishop et al., 1995] The structure of an artificial neural network is essentially composed of three parts:

- The input layer which is composed of artificial neurons, brings the initial data into the system for further processing by the following layers attached to it. Its purpose is to be the input to the ANN workflow.
- A hidden layer is located between the input and output layers of the ANN, in which the algorithm applies weights on the inputs and directs them to an activation function as output. A hidden layer can have a different number of artificial neurons than the input layer or the output layer and even hidden layers between them, there is no specific rule, or formula, to determine this but only from an experimental point of view. In other words, it allows to make non-linear transformation of the input data.
- The output layer produces the final result, it is at least composed of an artificial neuron. It receives the data from the previous layers, performs calculations to finally produce the output.

Backpropagation

The backpropagation technique uses the cost function noted L , which is often the MSE (Mean Squared Error) but there are others, which allow to evaluate the quality of the predictions, but also to calculate the error between the output produced by the network and the expected output, then uses this error to adjust the weights of the connections between neurons. This backpropagation process will propagate the error from the output layer to the input layer by using the derivation of the activation function to calculate the

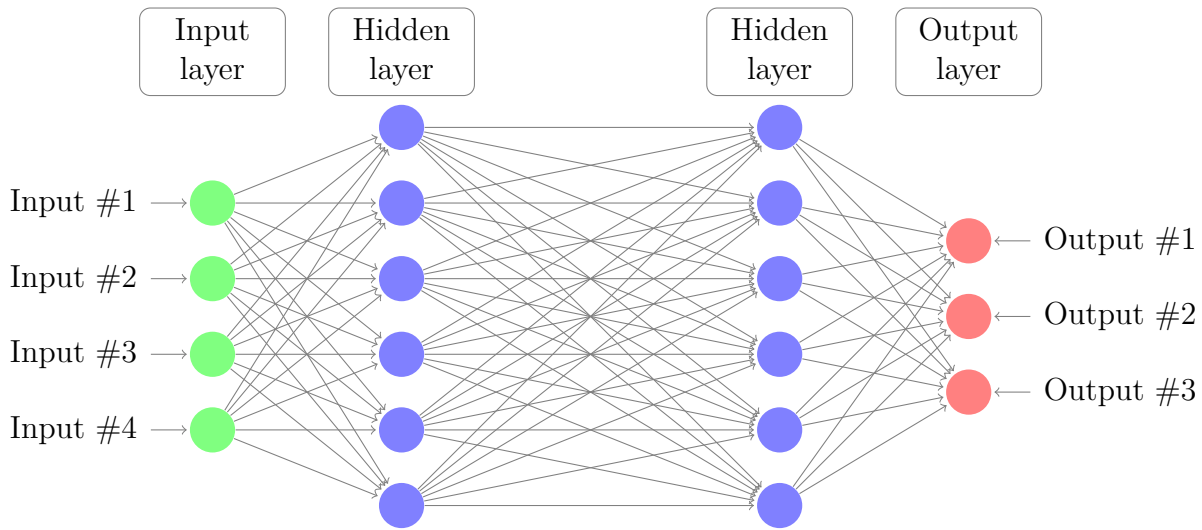


Figure 2.2 – Example of a multilayers perceptron

contribution of each neuron to this error.

Gradient descente

The objective is to minimize this cost function L , so that the prediction is more accurate by using a technique called gradient descent. The latter allows to calculate the partial derivative of the activation function by each interconnection weight to identify the gradient or in other words the magnitude of the change of weight necessary to reach the minimum of the cost function.

Thus to minimize the loss function L for a parameter of the model, i.e. a given weight, the algorithm starts from a random initial point, then the value of the slope of the curve is calculated at this point (by calculating the derivative of the function) $\frac{\partial L}{\partial w}$.

Here is the algorithm of the gradient descent :

$$W \leftarrow W - \alpha \frac{\partial L}{\partial W} \tag{2.9}$$

With the following parameters:

- W : the current weight value.
- α : the learning rate.
- $\frac{\partial L}{\partial w}$ the partial derivative of the loss function with respect to the current weight.

In the Figure 2.3, if our term $\frac{\partial L}{\partial w} < 0$, then our slope is negative and the algorithm will continue its iterative process by steps equal to the value α towards the right. On the

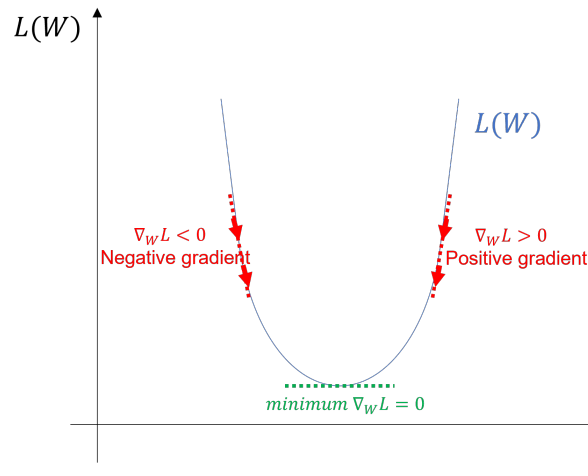


Figure 2.3 – Gradient descent

contrary, if our term $\frac{\partial L}{\partial w} > 0$ then the slope of the curve is positive, the algorithm will continue its iterative process towards the left until obtaining the global minimum. The choice of the parameter α remains important because if it is too big the value of the gradient will go from one point to another without reaching the global minimum and on the contrary if it is too small then the algorithm will put an infinite time and will never converge towards the global minimum.

2.2.4.2 Recurrent Neural Network (RNN)

Recurrent neural networks (RNN) are a particular family of neural networks adapted to sequential data and thus to signals of variable size. The input data noted x vary over time, will now be noted x_t with the index t being the position of the sliding window in the data sequence.

These RNNs are essentially based on two principles:

- The sliding window which allows to process signals of variable size.
- The recurrent connections allowing to analyze the past information of the input signal.

A recurrent connection makes it possible to reinject the output data in the input of this same layer, it also has a weight like all the other connections. These recurrent connections allow to see at the moment t the corresponding window but also to remember the previous decision.

Vanishing gradient

The major problem that remains on this kind of neural network is during the backpropagation phase through time (different from that of classical ANNs), more precisely on the use of the gradient descent algorithm. For this family of RNN when its architecture is larger, that the number of hidden layers or neurons increases, there appears a phenomenon of disappearance of the gradient during learning, the neural network is therefore difficult to train because the error tends to 0, the new values of the weights tend to the old values.

Exploding gradient

Conversely, these recurrent neural networks are also exposed to another phenomenon called gradient explosion which occurs when the derivatives of the slope become larger and larger as the backpropagation progresses. Thus the gradient does not converge and therefore will never find a global minimum.

In short, standard RNNs are not effective for long-interval dependencies due to gradient vanishing and gradient explosion [Hochreiter et al., 2001].

2.2.4.3 Long Short-Term Memory (LSTM)

LSTM networks are specific cases of recurrent neural networks capable of storing long-term dependencies introduced by [Hochreiter and Schmidhuber, 1997]. In order to model long term dependencies, it is mandatory to give the networks the ability to keep a state in memory over a long period of time, which is not possible with traditional recurrent neural networks. This memory capacity is realized by an internal memory called "cell" (noted c_t) in the form of a data vector that depends on the previous state noted c_{t-1} .

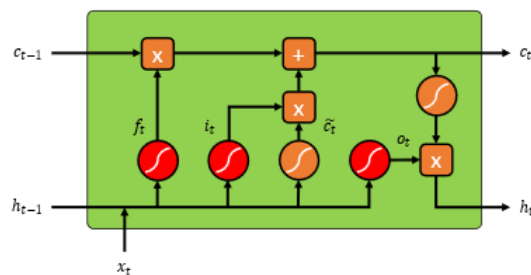


Figure 2.4 – LSTM Cell

The internal structure of the LSTM is composed of a forget gate (noted f_t), an input gate (noted i_t), an output gate (noted o_t) and the cell state (noted c_t). The inputs of these elements are weighted by weights and bias. This gives four weight matrices whose

dimensions depend on the dimensions of the previous output (noted h_{t-1}) and the input data vector (noted x_t). From the start, the input and previous hidden state vectors are concatenated. This new matrix M_{concat} will be distributed to all gates.

Forget Cell

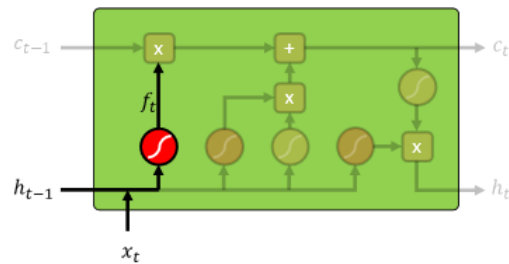


Figure 2.5 – Forget Cell

M_{concat} enters a first sigmoid function in order to have values between 0 and 1. With 0 which will allow to forget and 1 to keep the values. This output f_t will drive the multiplication function to let or not the values of c_{t-1} pass.

Input Cell

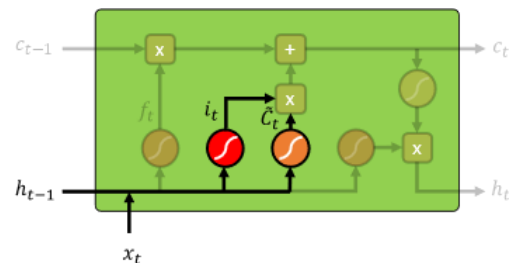


Figure 2.6 – Input Cell

M_{concat} goes through a sigmoid function so that the values are between 0 and 1. The value 0 means that the data is not important and conversely the value 1 means that it is important. At the same time M_{concat} enters a hyperbolic tangent function, the values of the matrix will be between -1 and 1 to help regulate the network (c_t). The following multiplication works like a valve and will be driven by the output of the sigmoid function in order to let or not the information that is important to keep from the output of the tanh function.

State Cell

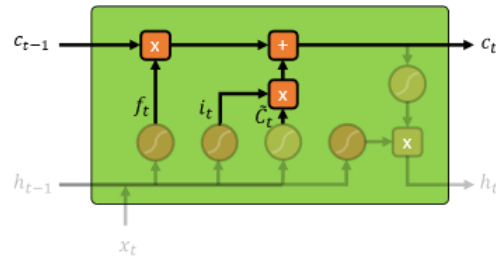


Figure 2.7 – State Cell

Now we have enough information to calculate c_t . The forgetting vector f_t was indeed multiplied by c_{t-1} with a possible loss in value. This is added with the result of the input cell to get the new value of c_t .

Output Cell

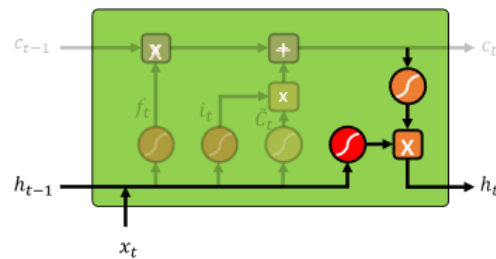


Figure 2.8 – Output Cell

Finally, we have the output gate that will decide what the next hidden state should be (which can be used for prediction). M_{concat} passes through the last sigmoid o_t and c_t passes through a tanh function. o_t drives the multiplication function to leave the information that h_t should keep.

In summary, the forgetting gate decides what is legitimate to keep from the previous steps. The input gate decides what information from the current step should be added. The exit gate decides what the next hidden state should be.

2.2.4.4 Gated recurrent Unit (GRU)

The Gated Recurrent Units (GRU), belonging to the family of recurrent neural networks, was introduced by [Cho et al., 2014] to solve the problem of vanishing gradient found in a RNN. It is considered as a variant of the LSTM in its operation but its internal architecture is different with an update and reinitialization gate .

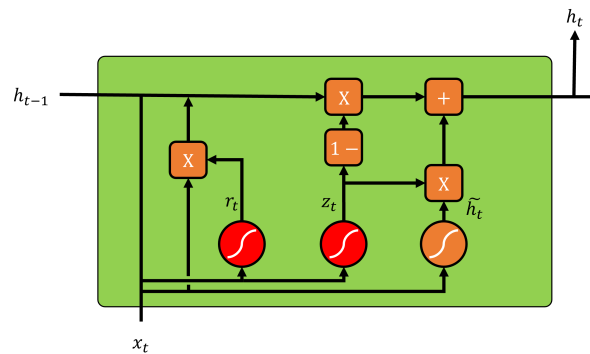


Figure 2.9 – GRU Unit

It is composed of three doors:

- Update gate, it works like the forget and input gates of the LSTM. It decides which information should be kept and which should be forgotten.
- Reset gate allows to control the number of past information to be forgotten.
- The state of the cell: the previous partially forgotten hidden state is combined with the input, which is then normalized by a hyperbolic tangent to have a value in -1 and 1. The GRU will then delete the previous data, judged useless thanks to the update gate, to add them to the previous hidden state h_{t-1} also judged useless in order to keep only the relevant information.

2.2.5 Evaluation metrics

In this section, we provide a slight overview of the evaluation methods used to assess the performance of a machine learning algorithm for predictive maintenance.

2.2.5.1 Confusion matrix

In supervised learning, the confusion matrix is a matrix that measures the quality of a classification model, and can therefore be used in the context of anomaly detection. Here is a representation of the confusion matrix:

		Predicted	
		Positive	Negative
Actual	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

This analysis depends on the starting hypothesis, in the case of anomaly detection, a normal behavior can be considered negative and an anomaly being considered positive.

- A True Positive (TP) is an anomaly data that the model has predicted as being one.
- A True Negative (TN) is normal data that the model has predicted as being normal.
- A False Positive (FP) is a normal data that the model has erroneously predicted as being an anomaly.
- A False Negative (FN) is a data of type anomaly that the model considers normal.

From the confusion matrix, we can deduce new evaluation methods for the algorithm:

- The precision measures the probability that an observation classified as positive is actually positive

$$P = \frac{TP}{(TP + FP)} \quad (2.10)$$

- Recall, allows to know the rate of true positives

$$R = \frac{TP}{(TP + FN)} \quad (2.11)$$

- F-measure, allows to know the performance of the model by taking into account both the precision and the recall.

$$P = \frac{2PR}{(P + R)} \quad (2.12)$$

2.2.5.2 Regression metrics

In this section we present a non-exhaustive list of metrics to evaluate a regression model:

- Mean Absolute Error (MAE) is an evaluation method that calculates the absolute difference between actual and predicted values, it is a measure of the average magnitude of errors in predictions, its formula is :

$$MAE = \left(\frac{1}{n}\right) \sum_{i=1}^n |y_{true} - y_{pred}| \quad (2.13)$$

- Mean squared error (MSE) is the average of the squared differences between predicted and actual values. It is a measure of average deviation between predictions and actual values. Its formula is :

$$MSE = \left(\frac{1}{n}\right) \sum_{i=1}^n (y_{true} - y_{pred})^2 \quad (2.14)$$

- Root mean square error (RMSE) provides a measure of the average deviation of the model predictions from the actual values. This method is often used to compare the performance of different regression models. When the predicted values are low, it means that the model predictions are close to the real values and therefore the model is more efficient. On the other hand this method is sensitive to outliers, so if there are too many of them, this method is less reliable. Its formula is :

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y_{true} - y_{pred})^2} \quad (2.15)$$

- R-squared is a statistical measure that represents the portion of the variance of the dependent variable explained by the regression model. The resulting value is between 0 and 1, the higher the value and therefore close to 1 then the model fits the data and conversely with a value close to 0. The formula is :

$$R^2 = 1 - \frac{\Sigma((y_{true} - y_{pred})^2)}{\Sigma((y_{true} - \text{mean}(y_{true}))^2)} \quad (2.16)$$

2.3 Anomalies detection in times series

In this section, we will address the problem of anomaly detection in time series. We will start with the types of anomalies treated in the literature. We will continue with several works using these techniques and concepts to finally discuss the results and the associated methods.

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [Chandola et al., 2009]. Anomaly detection is a problem that appears in a wide variety of fields such as astronomy [Lochner and Bassett, 2021][Henrion et al., 2013], medical [Tschuchnig and Gadermayr, 2021], transportation [Riveiro et al., 2017], finance [Ahmed et al., 2016] and many others. The use of machine learning to

detect anomalies in time series has been used in supervised and unsupervised learning.

Accurate and effective anomaly detection and diagnosis of modern engineering systems are crucial for ensuring reliability and safety and for maintaining desired product quality; most importantly detection may prevent disasters in the cases of equipment failure [Harrou et al., 2016].

2.3.1 Fields of application

In recent years, a lot of research has been done on anomaly detection in many different fields such as fraud detection, medical imaging, transportation, industry with failures, computer intrusions (cyber security).

- Fraud occurs when a group of individuals misuse resources provided by an organization, which provides a service, in an illegal manner for the sole purpose of gain. Among these illegal activities, we find credit card fraud with unusual purchases with high amounts. Smartphone fraud with fraudulent text messages, calls that when you answer you accept a service that will leave a high mobile bill. Car insurance fraud with document forgery and also insider trading in the stock market. It is in the interest of organizations to be interested in and implement these anomaly detection techniques to avoid large economic losses [Ahmed et al., 2016].

- Intrusion detection techniques are used to identify and prevent intrusions by malicious users who study vulnerabilities in computer systems or on computer networks. It is therefore essential for IT security to implement these intrusion detection techniques [Samrin and Vasumathi, 2017].

- The universe being infinitely large, the data which concern it are very numerous and do not cease increasing. New techniques of anomaly detection have been put in place to identify the most significant phenomena or space objects in order to prioritize their studies, which represents a real interest for the scientific community [Lochner and Bassett, 2021].

- In the medical field, anomaly detection techniques are used in the medical diagnosis of a patient from the data of devices such as scanners, which will allow to identify an abnormal condition of the patient, recording errors and even allow the tasks of classification of medical imaging, thus making the work of medical experts more simple and effective [Tschuchnig and Gadermayr, 2021].

2.3.2 Types of anomalies

Two important aspects of anomaly detection techniques are knowing that there is an anomaly but also determining the type of anomaly which, according to the literature, can be classified into three categories according to the literature [Chandola et al., 2009]:

2.3.2.1 Point anomalies

A data item that is isolated from all other data can be considered as a point anomaly.

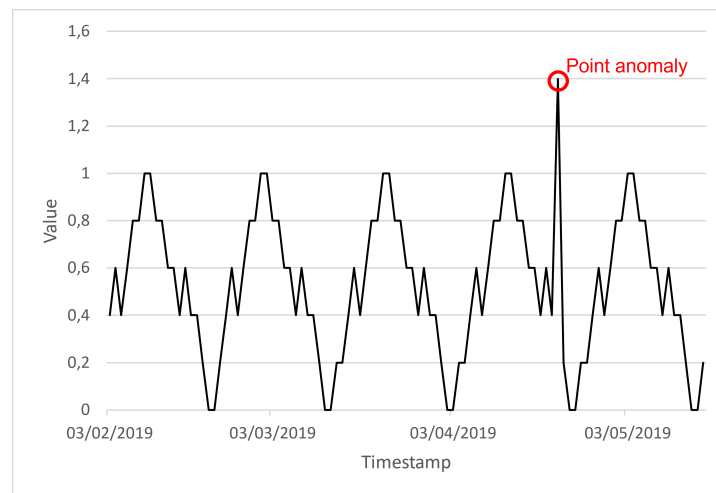


Figure 2.10 – Example of a point anomaly in a time series

2.3.2.2 Contextual anomalies

An instance of data that is anomalous in one specific context, but not in another, is called a contextual anomaly (which is also referred to as a conditional anomaly [Song et al., 2007]). Contextual anomalies are determined by combining contextual and behavioral characteristics. For contextual features, time and space are most often used, whereas behavioral features are behavioral characteristics depend on the domain of the study as average temperature, money spent or other quantitative measure.

2.3.2.3 Collective anomalies

A collection of data instances that looks abnormal in comparison to the whole data set is called collective anomalies. The individual data in a collective anomaly instance may not be anomalous in itself but the collection makes it anomalous.

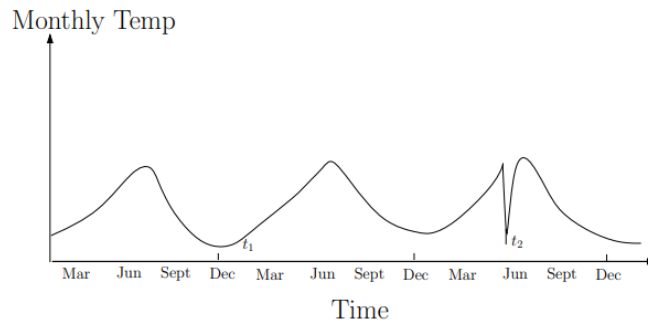


Figure 2.11 – Contextual anomaly in a monthly temperature time series (Chandola et al., 2009)

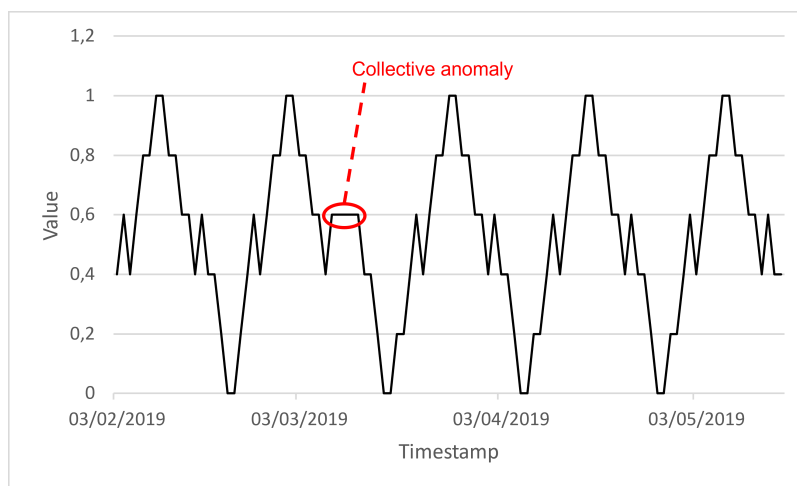


Figure 2.12 – Example of collective anomaly in time series

2.3.3 Machine learning-based anomaly detection

Considering the diversity of application fields and anomaly detection techniques in the literature, we have grouped different works in Figure 2.13 on knowledge base, statistics, clustering, spectral analysis, regression, pattern extraction, information theory and classification

2.3.3.1 Regression

Regression mentioned in section 2.2.3.1 can be used to estimate the difference between the actual value and the predicted value and if the difference, greater than a threshold, between the two is sufficiently large then this new data is considered as an anomaly. ARIMA (Autoregressive Integrated Moving Average) [Box et al., 2015] and SARIMA,

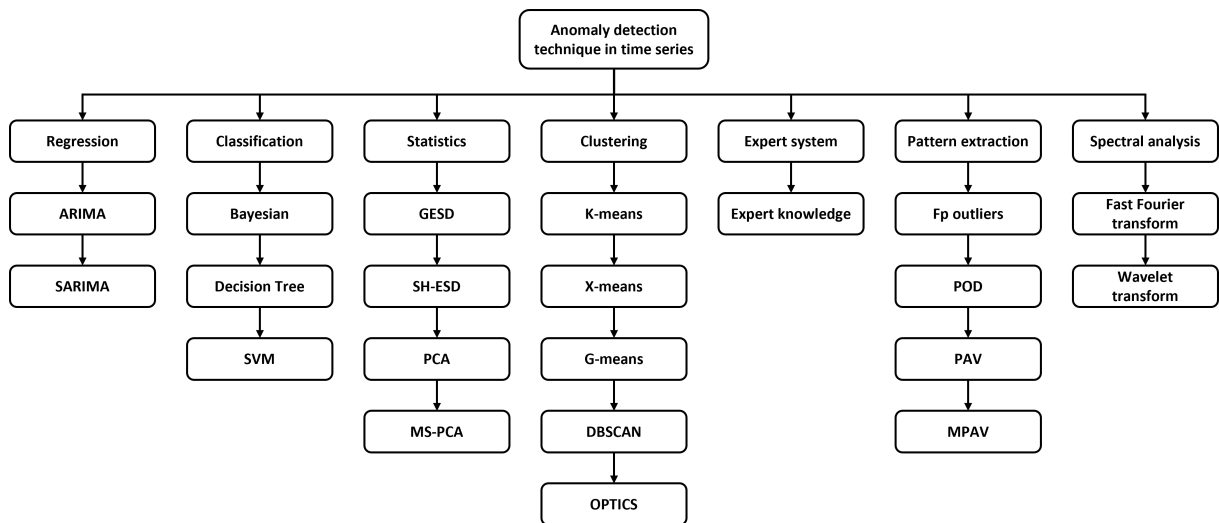


Figure 2.13 – Global deep learning techniques for anomaly detection in times series

are autoregressive models that can analyze or predict time series data and thus be used for the detection of anomalies such as: Additive Outlier, Innovation Outlier, Temporary Changes, Level Shift and Seasonal Level Shift [Chen and Liu, 1993].

2.3.3.2 Classification

- Decision tree learning mentioned in section 2.2.3.2, initially was not created for time series, but it can inherit some information from the original time series when used for example to make the understanding of a clustering problem easier [Ma et al., 2017]. A random forest is also a supervised learning algorithm that can be used for both regression and classification. Random forests create decision trees on randomly selected data samples and obtain a prediction for each tree and select the best solution by voting. [Sinwar and Kumar, 2016] This algorithm allows to obtain good results among others like SVR and KNN with the lowest false positives [Egaji et al., 2020].
- The SVM method was studied in section 2.2.3.2. [Dhiman et al., 2021] used a method based on an adaptive threshold and Twin support vector machine to identify anomalies on the temporal data of a wind turbine gearbox. Their choice allows a fast calculation and less sensitive to noise. This method was compared with other classifiers such as SVM, KNN, MLPNN, DT, the results show an accuracy of 95.75% on the temperature data of the gearbox bearings and an accuracy of 95.94%.

2.3.3.3 Statistical techniques

The GESD (Generalized Extreme Studentized Deviate) method, mentioned in section 2.2.3.3, only requires to specify an upper value for the probable number of anomalies present which can be problematic for some study cases and with an assumed normal distribution. To address this, Twitter [Hochenbaum et al., 2017] in 2017 created an algorithm named Seasonal Hybrid ESD that builds on the foundation of ESD, which detects both global and local anomalies. By employing a seasonal decomposition of the time series with also using the median and median absolute deviation (MAD) instead of the mean and standard deviation to calculate the z-score, as these are very sensitive to large numbers of anomalies.

Principal component analysis (PCA), mentioned in section 2.2.3.3, is one of them, as in the work of [Kini and Madakyaru, 2020] where Multi-Scale PCA (MS-PCA) is used as a modeling reference, combined with the GLR test based on a statistical hypothesis is used as an anomaly indicator. The results show a high fault detection rate and a low false alarm rate.

2.3.3.4 Clustering techniques

The clustering technique, mentioned in section 2.2.3.4, can be used for anomaly detection and these techniques can be summarized in three categories in the form of assumptions [Chandola et al., 2009] :

- 1st assumption: the data instances belong to a cluster included in the data while the anomalous data considered as anomalies do not belong to any cluster.
- 2nd assumption: Normal data instances are close to the centroid of their nearby cluster while anomalies are further away from the centroid of their nearby cluster.
- 3rd assumption: Normal data instances belong to large and dense clusters, while anomalies belong to small or sparse clusters.

In unsupervised learning, we find many clustering algorithms to implement anomaly detection such as K-Means, DBSCAN [Ester et al., 1996], OPTICS [Ankerst et al., 1999] and others...

2.3.3.5 Expert systems

Anomaly detection based on expert systems, mentioned in section 2.2.3.5, (also called "based on human knowledge of the domain") implies that the anomaly patterns are known and that it is possible to establish models understandable by the system. The knowledge is often represented by a set of rules defined by the system expert in the form of "if - then" which represent the different steps of the process, this list must be exhaustive. This method allows to have a wide range of knowledge and thus know-how to detect anomalies. However, the implementation of this kind of technique is costly in time and money and these same rules can change from one version to another of the expert system [Yairi et al., 2001].

2.3.3.6 Pattern extraction

There are essentially two parts: frequent pattern extraction and rare pattern extraction.

Anomaly detection approaches using frequent pattern extraction follow the example: if a dataset (instance) contains a lot of frequent patterns then it is unlikely that this data instance is an anomaly. The presence or absence of frequent patterns is then used to compute an anomaly score for each instance [Feremans et al., 2020]. Several algorithms have been used for anomaly detection based on frequent pattern extraction such as: FP-outliers (frequent pattern based outlier detection) [He et al., 2005], POD (Pattern-based Outlier Detection) [Zhang and Jin, 2011].

A similar approach has attracted the attention of the scientific community which is the search for rare patterns in time series because it would be more interesting than the previous approach. Thus, two algorithms have been proposed [yun Chen and yan Zhan, 2008]: PAV which is an anomaly detection algorithm based on infrequent patterns has the advantage that it does not need any training to directly identify an anomaly. By combining this algorithm with the Haar Wavelet they obtained a new algorithm MPAV which allows this time to detect anomalies at different scales by using the multi resolution property of wavelets.

2.3.3.7 Spectral analysis

Spectral analysis, mentioned in section 2.2.3.7, is used in [Han et al., 2014], the authors use the Fourier transform to represent the signals in the frequency domain and study the vector similarity between the temporal signals. This allows an efficient comparison of the frequency characteristics of signals for similar events. Thus, a non-similar event can be considered as an anomaly. Another approach by [Park et al., 2021] consists of a new method called the partial Fourier transform which is based on the traditional Fourier transform, but which is faster and more accurate than the latter. This approach consists in keeping only the most significant parts of the frequency data.

The wavelet transform also allows to extract significant information from a wave and to be used for the detection of anomalies [Lu and Ghorbani, 2008] and [Du et al., 2018].

2.3.4 Deep learning-based anomaly detection

Deep learning is a branch of machine learning that allows to try, from raw data inputs, to model complex problems from non-linear articulated architectures in an automatic way. In this section, we will study how anomaly detection techniques are used from deep learning, based on artificial neural networks.

2.3.4.1 Artificial Neural Network (ANN)

This work, performed by [Pliego Marugán et al., 2019], shows that detecting false alarms in a wind turbine increases its reliability. They used a new approach with artificial neural networks to detect false alarms and to prioritize them. The artificial neural network architecture consists of three different multilayer perceptrons to analyze data from a SCADA and condition monitoring system. They also filtered and used pattern recognition to analyze the dataset from the European OPTIMUS project, this pattern recognition takes into account the history to train the neural network. The results show an accuracy and a sensitivity higher than 80% and in other cases higher than 90%, they also compared these results with another method called fuzzy logic which also gives good results, but below the neural network based architecture.

2.3.4.2 Long Short-Therm Memory (LSTM)

In this section, we have summarized in a table different works that deal with anomaly detection in time series using deep learning tools and methods, especially using the Long Short-Therm Memory (LSTM) architecture, mentioned in section 2.2.4.3.

[Park et al., 2018] introduce a long short-term memory-based variational autoencoder (LSTM-VAE) that fuses signals and reconstructs their expected distribution by introducing a progress-based varying prior (This refers to an a priori probability distribution that assigns higher probabilities to outcomes that show progress or improvement over time, while taking into account the evolution of this progress over time). Their LSTM-VAE-based detector reports an anomaly when a reconstruction-based anomaly score is higher than a statebased threshold. For evaluations with 1555 robot-assisted feeding executions, including 12 representative types of anomalies, their detector had a higher area under the receiver operating characteristic curve of 0.8710 than 5 other baseline detectors from the literature.

In this article [Lindemann et al., 2020] presents a novel detection and prediction procedure based on a LSTM architecture to cooperatively predict process outputs and anomalies by using two separates but interacting models. The anomaly detector is realized as stacked LSTM autoencoder and the cooperative prediction models are based on sequence-to-sequence networks (Seq2seq), is a model that takes a sequence of items and outputs another sequence of items (example: translator), with gated recurrent units for short-term and LSTM for long-term effects. The approach is evaluated within a real industrial environment by means of a production plant for hot forging at a German automotive supplier for metal components. The results show a decent detection ability of the anomaly detector for both abrupt and slinking anomalies. The stacked GRU approach outperforms the Seq2Seq LSTM approach for short-term dependencies and the LSTM the GRU for long-term anomalies. The cooperative prediction increases the accuracy and is transferable to different process types.

[Ding et al., 2020] describes the application of an LSTM-based error detector for a system of two industrial robotic manipulators. A detailed Simulink model provides the training data for an LSTM predictor. Error detection is achieved via intelligent processing of the residual between the original signal and the LSTM prediction using two

methods. The first method is based on the non-parametric dynamic thresholding. The second method exploits the Gaussian distribution of the residual. The parameters of the LSTM model are Lookback is 250 steps. Lookahead is 10 steps, two consecutive hidden recurrent layers are fully connected with a dropout of 0.3 in-between and both layers consist of 80 LSTM units. The paper demonstrates that such LSTM based error detection method can be created and applied in model-based engineering in an automated manner.

The study conducted by [Malhotra et al., 2015] suggests using stacked LSTM networks for anomaly/fault detection in time series. A network is trained on non-anomalous data and used as a predictor over a number of time steps. The resulting prediction errors are modeled as a multivariate Gaussian distribution, which is used to assess the likelihood of anomalous behavior. The architecture of the model is made of two layers with 30 units and 20 units respectively. For ‘space shuttle’ and ‘power demand’ datasets which have long-term temporal dependencies along with short-term dependencies, LSTM-AD (LSTM- Anomaly Detection) shows significant improvement of 18% and 30% respectively over RNN-AD in terms of F0.1-score.

[Zhang et al., 2022] proposes an anomaly detection and diagnosis method for wind turbines using LSTM-SDAE (LSTM Stacked Denoising AutoEncoder) and XGBoost. An abnormal data recognition algorithm based on LOF and adaptive K-means was first developed for data preprocessing and noise extraction. The LSTM-SDAE model was then established to learn the spatial correlation information between several different variables and the temporal characteristics of each variable in normal behavior modes. The Mahalanobis distance was calculated based on reconstruction errors and the threshold for anomaly detection was set using a 99.7% confidence interval for the distribution curve fitted by KDE. An alarm mechanism based on a sliding window was set up to detect abnormalities in real time. Finally, contribution analysis was conducted to extract the parameter features under different abnormal modes, and the XGBoost was trained by extended data from wind turbines of the same type in the same wind farm to realize anomaly location and diagnosis. Specific examples from a wind farm located in north-eastern China were considered. It is evident that the proposed LSTM-SDAE achieves a better performance than the LSTM-SAE, SDAE, and SAE with less root mean square error (RMSE) and mean absolute error (MAE) for the testing set by considering the spatiotemporal nonlinear relation and model generalization.

Table 2.2 – Overview of the different articles on anomaly detection with LSTM

Source	Input		InputModel		Evaluation		
	Data Type	Labels	Anomaly types	Architecture	Scenario	Metrics	Performance
[Park et al., 2018]	Multivariate time series	No	Outlier contextual	Variational LSTM AE	Environ, anomalies in robots systems	AUC, ROC, recons, error	Higher than HMM, SVM, AE
[Lindemann et al., 2020]	Multivariate time series	No	Outlier, collective contextual	Observer-based LSTM AE	Discrete manufacturing	Recons. Error	No comparison
[Ding et al., 2020]	Multivariate time series	No	Contextual	LSTM +EWMA	Industrial robotic manipulators	Precision recall	No comparison
[Malhotra et al., 2015]	Multivariate time series	No	Contextual	Stacked LSTM	Several such as power demand	Precision, Recall, F1-score	Higher than RNN
[Zhang et al., 2022]	Multivariate time series	Yes	Outlier, contextual	LSTM-based stacked denoising AE, XGBoost	Diagnosis wind turbines	Accuracy	Multiple classifiers but XG-Boost more accurate but not the fastest

2.3.4.3 Gated Recurrent Unit (GRU)

In this section, we have summarized in a table different works that deal with anomaly detection in time series using deep learning tools and methods, especially using the Gated Recurrent Unit (GRU) architecture, mentioned in section 2.2.4.4.

[Lee et al., 2018] propose a model with S-CNNs (Stacked Convolutional Neural Networks) and a S-GRUs (Stacked Gated Recurrent Units). The S-CNNs layers firstly capture spatial feature extraction of the input sequence data of vibration sensor, and the result is used to temporal feature learning secondly with the S-GRUs. After this procedure, finally a regression layer predicts an anomaly detection. Experiments show that hybrid architectures with S-CNNs combined with S-GRUs provide better results than GRU-only architectures with 7.18% RMSE on average.

[Chen et al., 2020] present a data-driven integrated framework for anomaly detection in coal pulverizing systems, comprising of two modules - condition value prediction and anomaly detection. The framework uses a GRU model for condition value prediction and a novel unsupervised clustering algorithm for anomaly detection. The proposed framework is evaluated on an industrial coal pulverizing system and shows promising results. However, the GRU model is not optimized, with a structure of GRU(50)–GRU(50)–Dense(20)–Dense(1) but better than LSTM(50)–LSTM(50)–Dense(20)–Dense(1). and NN has the

structures of 50–100–50–1, and the proposed clustering algorithm has a slow degradation process. Future work should focus on optimizing the GRU model for improved accuracy in temporal information extraction and improving the accuracy of the proposed clustering algorithm for early anomaly detection and its application to other degradation processes.

Table 2.3 – Overview of the different articles on anomaly detection with LSTM

Source	Input		InputModel		Evaluation		
	Data Type	Labels	Anomaly types	Architecture	Scenario	Metrics	Performance
[Lee et al., 2018]	Multivariate time series	Yes	Contextual	Stacked GRU Stacked CNN	Bearing rotating machinery	MSE, RMSE, MAE	No comparison with other methods carried out
[Chen et al., 2020]	Multivariate time series	No	Contextual	GRU with clustering	Coal pulverization system	Prediction error	Proposed clustering algorithm better than others

2.4 Conclusion

In this chapter, we have outlined the main tools and methods needed to understand the context and the problem we address in this thesis. We have therefore exposed the different programming tools, programming libraries, cloud solutions as well as machine learning methods and the means to evaluate them to apply predictive maintenance. We continued with the definition and types of existing anomalies, the application domains, the machine learning methods used in the literature. Thus, for our experiments, we chose to use Python, because the user community is larger and the development is simpler. To launch our calculations, Dynavia provided us with access to a NaaS from OVH. Then, we chose to use LSTM recurrent neural networks, because according to the literature, they are able to model temporal dependencies, to model long temporal data sequences, to handle noisy sequential data efficiently and to cope with irregularities in the data. In the following section, we will present our contributions concerning a predictive maintenance strategy applied in an industrial context.

PREDICTIVE MAINTENANCE STRATEGY APPLIED IN AN INDUSTRIAL CONTEXT

3.1 Introduction

This chapter is mainly dedicated to a proposal of a global predictive maintenance strategy method by describing each step, with the associated methods and experimental tools, implemented throughout the thesis.

To implement a predictive maintenance strategy, we propose the workflow (Figure 3.1) which summarises all the main steps to be taken to achieve this objective.

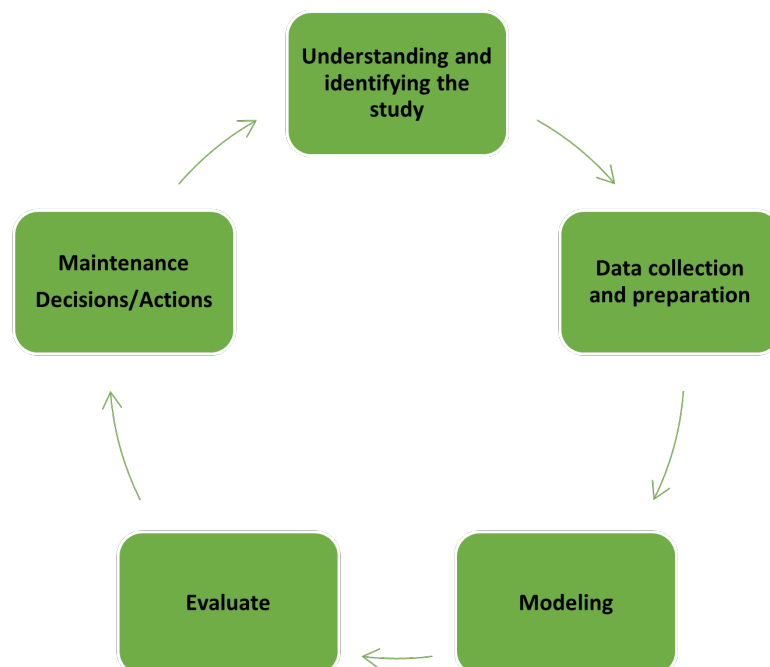


Figure 3.1 – Predictive maintenance strategy workflow inspired from [Achouch et al., 2022]

The first step is to understand the functioning of the system in its entirety, thanks in particular to the functional analysis. Once the global vision is established, it is necessary to find the candidate(s) for predictive maintenance. For this purpose, we propose to use a set of methods such as the calculation of Overall Equipment Effectiveness (OEE, mentioned in more details in section 3.2.3) and the global and group availability combined with a Pareto diagram in order to identify the parts of the system providing functions. The research will be further strengthened by analyzing failures and failure rates to identify the most prone components, and the selection will be complemented by an analysis of maintenance history.

The second step is the collection, the preparation of the data. This step will consist of extracting the data needed for the study and applying various treatments to prepare them for the modelling step.

The modelling stage creates a model from the input data for expected outputs. This step includes the choice of the algorithm whether it is regression, classification or clustering, the choice of parameters and its training.

The next step is to evaluate the model with the evaluation metrics, to prove that it works, that it describes the data well and that it is generic. Once this is done, just deploy and maintain it.

The last step is to interpret the results of the model and different performance indicators to identify possible scenarios and make decisions to implement maintenance actions.

3.2 Understanding and identifying the study

This funnel method (Figure 3.2) establishes a coherent analysis by starting with a global analysis of the system with the objective of reaching a target, a potential candidate for predictive maintenance, using analyses.

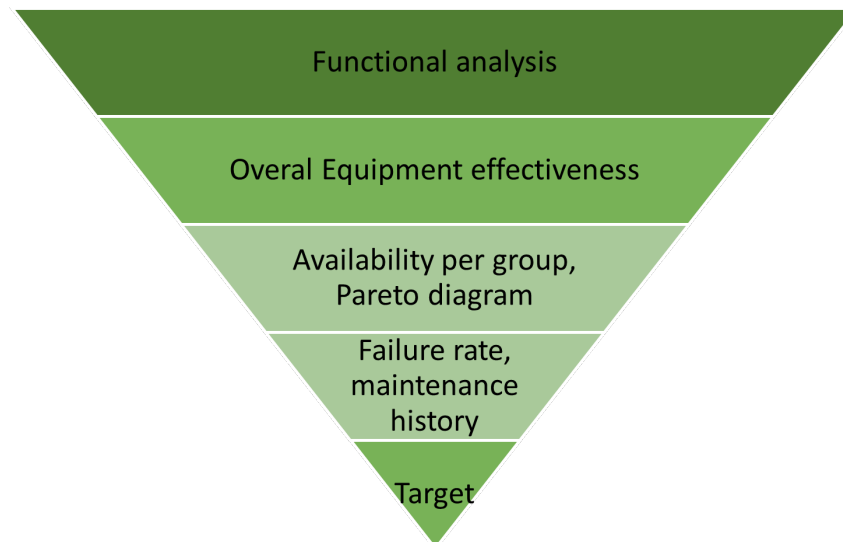


Figure 3.2 – Analysis funnel to identify a candidate for predictive maintenance

3.2.1 Functional analysis

Functional analysis is a method applied during the design of a product or system. It enables a solution to be built in relation to the customer’s needs by means of a set of functions and not by a technical solution from the outset, because a technical solution can evolve over time or even disappear. As for thinking in terms of function, it does not change over time.

Many manufacturers make the mistake of responding to the customer’s need with a technical solution from the outset and find themselves in difficulty when they realise that the solution does not meet the full function required.

As we said, its origin is the customer’s need, which must be expressed clearly, precisely and unambiguously. If one of these criteria is not met then the need is poorly formulated, resulting in poor specifications, poorly thought out solutions and often ending in litigation

between the parties.

The objectives are:

- Identify all the functions of the system in its different phases of use.
- Describe a hierarchical breakdown of these functions.
- Describe the composition of each function.
- Define the interactions between each function.
- Define the interfaces with other systems and the environment.

In predictive maintenance, functional analysis is used to understand each function of the system under study, but it also provides a context for the study. There are several methods that allow to represent the system under several functions such as :

- The APTE method, also known as the "horned beast", which puts the system at the centre of the representation with interactions between different external environments.
- The FAST method (Function Analyses System Technique) which is a block representation with different cross interactions to different blocks to answer the questions: Why this function? When? How?

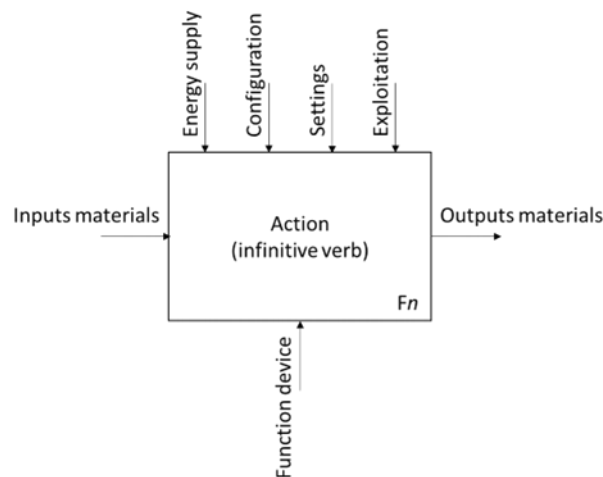


Figure 3.3 – SADT Black box

The Structured Analysis and Design Technique (SADT) representation method (Figure 3.3), also known as IDEF0 (Integration Definition for Function Modeling), was devel-

oped for Softech by [Ross and Schoman, 1977]. It allows a function to be represented in a block with the following characteristics:

- Action: this is the function that is performed by the product to meet the customer's need.
- Function device: this is the component provided to the user to meet the need.
- Inputs/outputs materials: this is what the product acts on, it can be energy, information, material.
- Control constraints: These are the parameters that trigger or modify the execution of a function.

The function blocks are then interlocked with each other (Figure 3.4) to build a whole functional system.

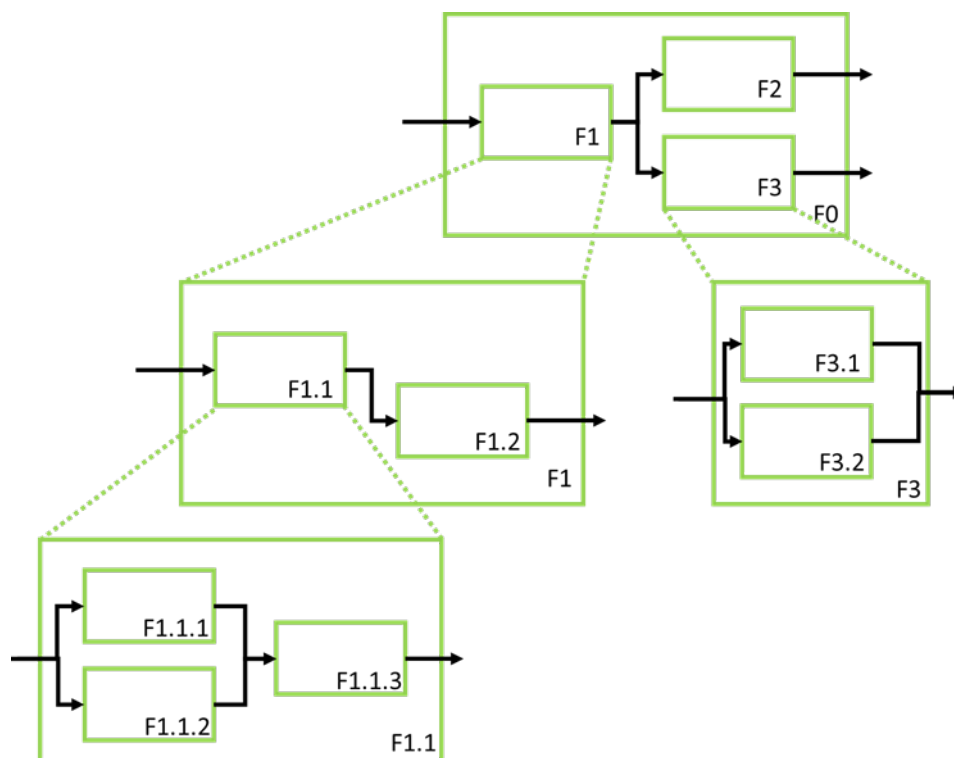


Figure 3.4 – Example of a SADT model by level

Thus, a functional block of an entire chain can be broken down into several levels. Level 0 represents the system as a whole, represented here by block F0, which itself contains n blocks, in this case 3. Then a block of level 1 can also be broken down into several

n blocks and so on.

The challenge is to decide on the granularity of the functional breakdown, because if we are global, we risk missing the key element of our study, and conversely, if we are too precise, we risk ending up with many elements to study. It is therefore necessary first to concentrate on a global study, to carry out analyses, and finally to prioritise the studies. Then, depending on the constraints of the study and the difficulties, it may or may not be possible to deepen the previous granularity. It is also necessary to have business (process) and technological expertise available to understand the functioning of the system, its processes, its operating mechanisms and the hardware.

Applying the previous precepts to the case of the Dynavia system, we obtain the representation 3.5. With F0 the whole system, F1 being the management of the formulated products, which can be decomposed into two sub-functions of storage and dosing. Where the output of block F1 is the input of block F2. The latter allows the mixing of the formulated products.

Block F1.1 represents the Storage formulated products, the input is the formulated product in a product container (IBC) which is subsequently stored in a stainless steel tank through various controls, including electrical power, pneumatic power, controls and tank selection.

In block F1.2 the stored product will be dosed. This time the process is done by a pre-selected dosing pump. The pump will inject the required amount of product for the production.

Block F2.1 is in another area, which mixes all the formulated products selected for the current recipe and then provides a transfer.

3.2.2 Early planning for confidentiality and access to data in research studies

Before any analysis can be carried out, it is essential to have a confidentiality and data use agreement in place. If you are a manufacturer and you sell a system, then the data generated by the customer does not belong to you, if you want to exploit it to optimise

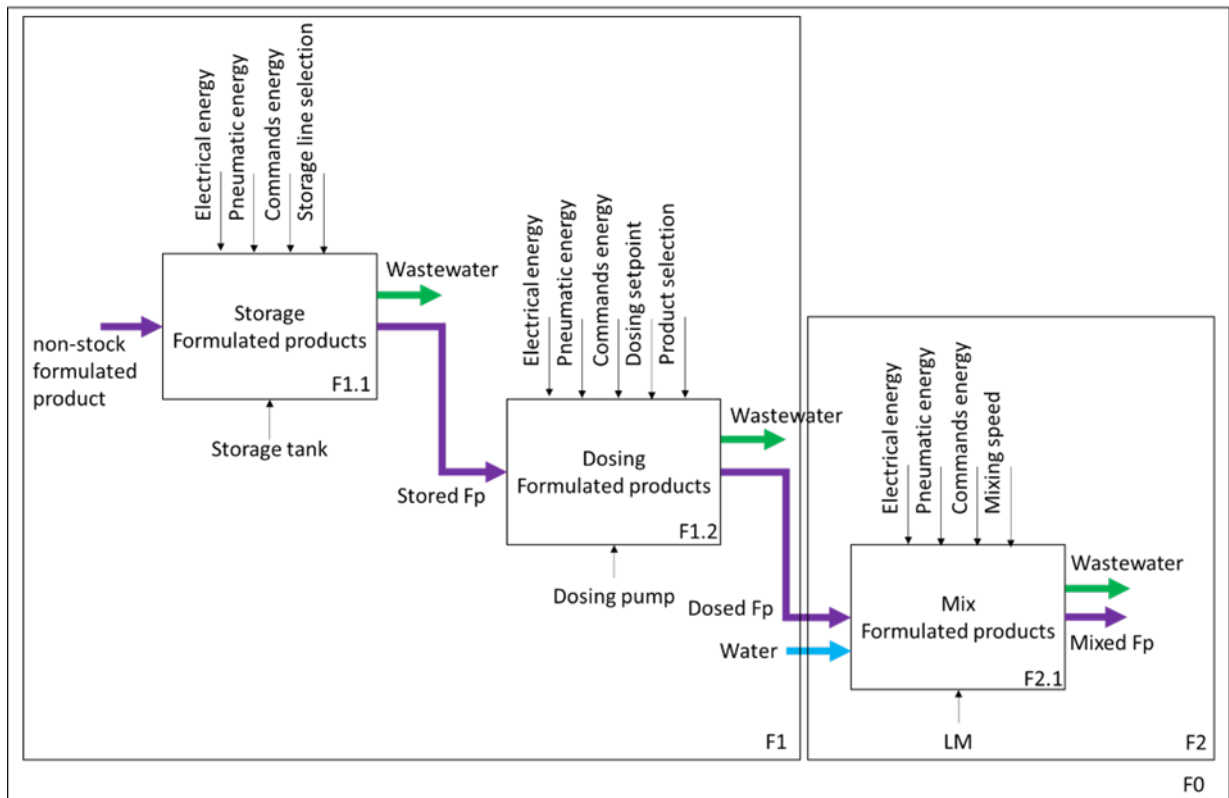


Figure 3.5 – Extract from a Dynavia function block assembly

your system and make it more efficient, then it is in your interest and in his interest to put in place the confidentiality agreement for the exploitation of the data and that he remains the owner of it.

But you may be faced with restrictions for confidentiality and security reasons, in which case you need to express your need for accreditation to access this data specifically for your study and to have the means to work with it in complete confidentiality.

This kind of administrative procedure can take more or less time depending on the size of the company and the negotiation and validation process that must be put in place. It is therefore advisable to plan ahead and to start these steps at the beginning of the study or even before.

3.2.3 Overall Equipment Effectiveness (OEE)

In this section, we focus on the key performance indicators (KPIs) for manufacturing relevant to our work, which are mostly defined in the ISO 22400-2 standard [International Organization for Standardization, 2014], Automation systems and integration - Key performance indicators for manufacturing operations management - Part 2: Definitions and descriptions. The list of KPIs is numerous, it is important to define which ones are necessary for the study, depending on the context of use. It is also important to have all the necessary data available to implement them.

For a manufacturer and for a user, the KPIs will not be the same, because the manufacturer will focus solely on the performance of the system it manufactures, because they will be marketing arguments for sales, but also a means of positioning itself in relation to the competition, but in a more or less precise context and conditions of use. Whereas for a user, these KPIs will initially serve to know whether or not he will buy the equipment according to his needs. However, once the system is deployed across an entire production line, the number of key performance indicators (KPIs) will increase, and the KPI values presented by the manufacturer are likely to differ. This is because the interconnections between systems can disrupt the upstream or downstream performance of each system.

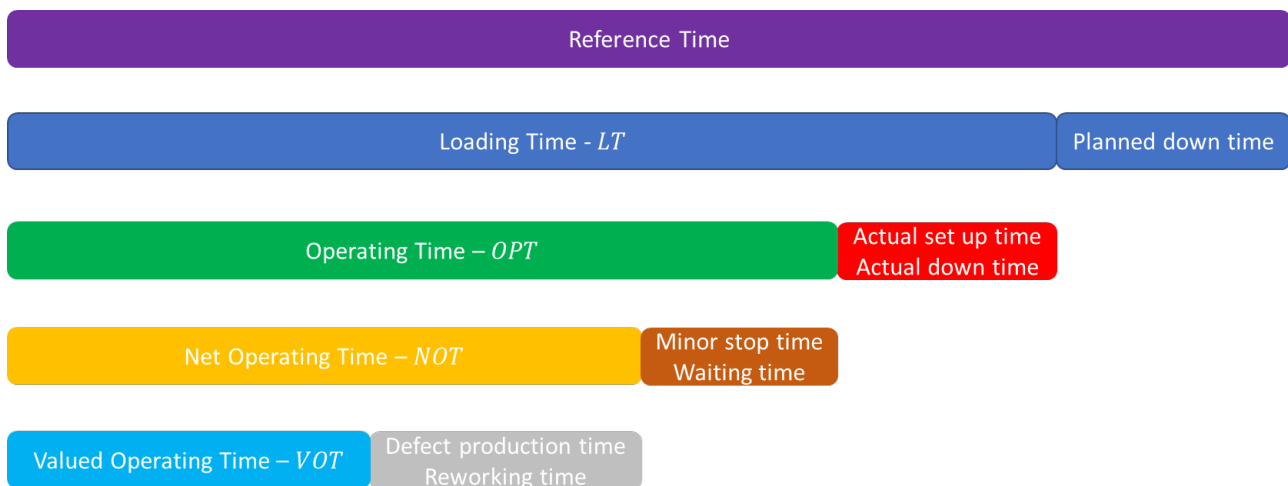


Figure 3.6 – OEE from ISO 22400-2 standard

OEE is a measure of the efficiency of a machine park, one or more installations, or an entire production line. The OEE index forms the basis for improvements by providing information on production, identifying various production losses and improving the quality

of the end product through optimised processes. It is defined by the availability used, the efficiency of the work unit and a finished product rate:

$$\text{OEE index} = \text{Availability} \times \text{Performance rate} \times \text{Finished goods ratio} \quad (3.1)$$

It is expressed as a percentage ranging from 0 to 100, which can be measured on demand or at fixed periods.

Availability

Availability indicates the proportion of time the equipment is actually in use, Operating Time (OPT) compared to the Loading Time (LT). The availability represents the extent of the loss due to equipment downtime.

$$\frac{OPT}{LT} \quad (3.2)$$

With loading time equal to reference time minus planned downtime and operating time which is loading time minus actual start-up time and actual downtime. It is measured as a percentage, ranging from 0 to 100.

Performance rate

The performance ratio is the ratio between the net uptime and the uptime.

$$\frac{NOT}{OPT} \quad (3.3)$$

It represents the difference between the target cycle time and the actual cycle time.

Quality ratio

The quality ratio is the ratio between the quantity of goods (GQ) and the quantity produced (PQ). Measured as a percentage ranging from 0 to 100.

$$\frac{GQ}{PQ} \quad (3.4)$$

3.2.4 Unavailability per groups and Pareto diagram

3.2.4.1 Unavailability per groups

Previously, we determined the overall availability of our system in order to have a starting indicator to measure the efficiency. We are now interested in determining which area of the equipment shows the most unavailability during production. Both machine downtime and logistics time must be taken into account. In order to do this, it is necessary to identify the data in the different data sources:

1) CMMS

The maintenance department will also provide you with a copy of their CMMS where you will find the details of each maintenance team intervention on the system. Of course, different CMMS software or other maintenance monitoring methods have different formats, but it is necessary to have a minimum of information to exploit key performance indicators and identify redundant failures.

The different stages of maintenance

The objective of a manufacturer is to produce in optimal conditions with reliable equipment with the best performance. But prolonged use, or conditions of use that are not respected, can lead to the failure of a part of the system, resulting in production stoppages if the latter is essential. A failure can be partial or total, in other words, a failure means that a system no longer fulfils the function for which it was created. The maintenance team must intervene as quickly as possible to put the equipment back into service and restart production. It is necessary to define the different maintenance times and to record them at each intervention in order to optimise them. Figure 3.7 shows the different maintenance times:

MTTF (Mean Time To Failure)

The average length of time a system has been operating before the first failure. This basic measure of reliability is used for non-repairable systems (one shot). It represents the expected duration of operation of an element until it fails (e.g. a fuse). Although this measure is close to MTBF, they should not be confused.

MDT (Mean Down Time)

Average downtime which is the sum of the time to detect the failure with the time to

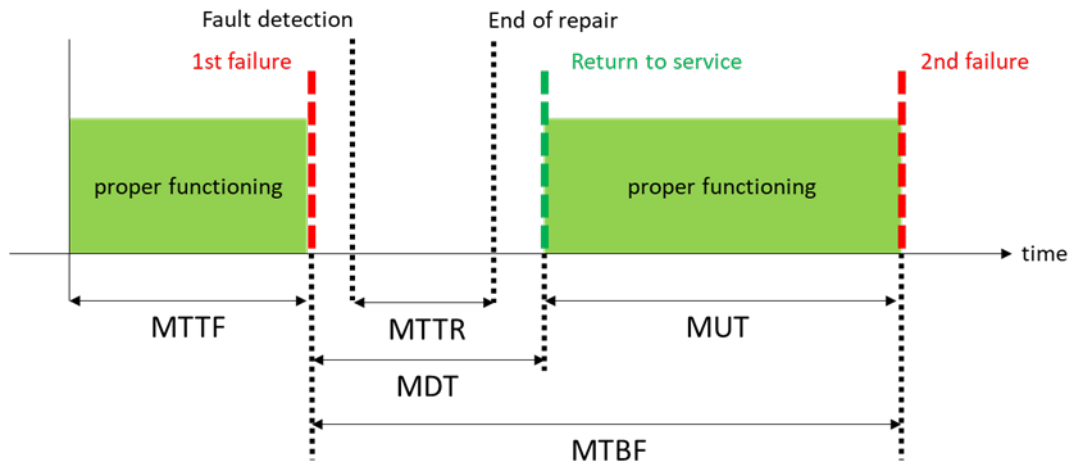


Figure 3.7 – The different stages of maintenance

repair and the time to get back into service.

MTTR (Mean Time To Repair)

Average time to repair. Taking too long to repair a system is not desirable for a manufacturer, as it can have a negative impact on business results with delays in lead times and lost revenue. Understanding and properly addressing the MTTR is an important tool for any organisation, as it indicates how effectively the manufacturer can respond to and repair any system problems with its assets. The aim of organisations is of course to reduce MTTR with an in-house maintenance team supported by the necessary knowledge, resources, tools, spare parts and CMMS software.

MUT (Mean Up Time)

This is the mean time to completion or mean time to availability of a system. In other words, it is the average time of operation between the last return to service after repair and the next failure.

MTBF (Mean Time Between Failure)

The average time between two consecutive failures. This measure is used to estimate the time that elapses between two unplanned failures. It is used for repairable systems, but its calculation does not take into account units that are shut down for scheduled maintenance (servicing, lubrication, recalibration, etc.) or the preventive replacement of spare

parts. The MTBF does not consider planned maintenance, but by its value, it allows the manufacturer to predict the frequency of inspections of the system even if it will probably operate for a certain number of hours before the next failure. It can then implement preventive actions to help keep the system running smoothly and thus extend the life of the asset.

Here is an example of a maintenance history:

The creation of a maintenance history necessary to establish predictive maintenance, so it takes discipline to fill it in correctly. The number of columns is not exhaustive, as the "invoicing" column can be deleted or replaced if the customer is autonomous and does not have an after sales service with the manufacturer.

- The date and call time columns correspond to the date and time when the failure occurs and the maintenance service is called.
- Operator, is the person requesting assistance.
- The technician, is the person who intervened for the maintenance action.
- The beginning of the intervention is a concept that can be different from one company to another and it must be defined with each party. In one case, it may be the moment when a remote maintenance technician begins to intervene, diagnose, manoeuvre until the system is back in service, and in another case, it is the moment when, after a call, he travels to the area in question, he intervenes. In the first case, the call time will be equal to the intervention time, but in the second case the time will be different.
- The time of the end of the intervention also depends on the definition of the parties. In the first case, the call time will be equal to the intervention time, but in the second case the time will be different. Let's illustrate with an example: a maintenance technician has just finished repairing a hydraulic pump, but he has to return to the supervision area to restart production.
- The zone column is the location of the fault, even if it is a computer programming fault, then it is sufficient to quote the location of the computer equipment.
- Tag is the column that will give the identifier of the component that has failed, which is often a code defined by the user company to identify its equipment.

- Type, allows you to know what type of failure was: mechanical, electrical, programming, human error... It is possible to enrich it with another column to give an additional but succinct description. For example: a hydraulic pump is faulty, type = electrical; a cable fault.
- The reason for the call, allows you to recontextualise the reasons for the request for maintenance with the initial diagnostic information and to know what was done to arrive at this situation.
- The resolution, allows to describe all the diagnostic steps, the resources used, both material and human, to put the system back into service.
- The ticket column, is the number of the intervention in order to list this intervention and to find it more easily.
- Blocked machine, is the system entirely faulty and unable to produce or only a part of an area is faulty, but needs to be put back into service for later, because the level of criticality and urgency are no longer the same. If there is a maintenance contract between the operator and the service department, it may be an argument for billing, as the urgency is not present and a maintenance ticket would have been sufficient to deal with the problem.
- The billing column determines whether there will be billing for the maintenance intervention, if the service is outsourced. Generally, the criterion is whether or not the customer would have been able to troubleshoot on their own. A second criterion may be that there is no billing, because the failure and resolution allowed the manufacturer to redesign its system and make it more reliable (e.g. software process update), if it is fair.

Also make sure that the information technology department gives you access to the data documents, as in some cases the values of certain data such as PLC grafcet represent process steps to have the description of that step.

2) System alarms

The classification of the alarms makes it possible to find the unavailability of the system. These data correspond to pauses, upstream waiting, downstream waiting, blockages... These are often found in the form of grafcet alarms on the PLC.

Here again, it is possible to encounter several problems:

- The choice of alarms.
- The overlapping of alarms, between them. Because an alarm can trigger several of them.
- Overlapping of alarms over several days.
- The same Distributed Control System (DCS) or Supervisory Control And Data Acquisition (SCADA) system managing several production lines at the same time. It will be vital to separate the alarms and therefore the unavailability by production line.

Once these problems have been dealt with using the programming language of your choice, it is sufficient to add up the times of the alarms by group to find out which zone is the most unavailable.

Here is an example with the Dynavia case:

- Let's start with the choice of alarms.

Table 3.1 – Extract alarm classification - Dynavia

Priority	Category types	Color	Alarm reset type	Alarm type	Messages
0	leak	red	Mandatory	Liquid leakage on machine	Product leak on... detected !!!
5	G7	Deep blue	Not mandatory	G7 blockades in progress	
6				G7 SFCReset blocking	
7				G7 Dcy blocking	
9				Dosing error	
10			Not mandatory	Machine in pause	
24				Expectations Up-stream machine	
25				Dynavia waiting	
26				Waiting Down-stream Machine	

This Table 3.1 is the classification of alarms given by the IT department. It shows the

priority of triggering, and therefore of resolution, the type of category (process, security, energy, network, etc.), the associated colour for the HMI, whether or not there is a need to reset the system, the type of alarm and the message that follows. We advise you to consult with the IT department to choose the right alarms for your study. If you are the customer using the system in a production line, you will be interested in alarms 24 and 26, respectively the upstream and downstream expectations of the machine, whereas if you are the manufacturer, you would not select them. We will select alarms 5, 6, 7, 9, 10 and 25.

- Overlapping alarms

Figure 3.8 shows a representation of the alarm data, highlighting the overlapping timestamps shown in blue and red. The first column shows all timestamps, with the start of the alarms circled in red and the end of the associated alarms circled in blue. The second column shows the identifiers, while the third column shows the duration of each alarm. It is important to note that there are duplicates in the third column, as this is clearly visible.

21/10/2019	10:41:37.792040	ALARM:DYN004/5/PremixMA/M/ Premix in pause : ATMX not ready	4.680
21/10/2019	10:52:15.554417	ALARM:DYN004/5/PremixMA/M/ Transfer premix WBIX in pause : ATMX not ready	1.239
21/10/2019	10:52:15.554424	ALARM:DYN004/5/PremixMA/M/ Premix in pause : ATMX not ready	1.239
21/10/2019	10:52:15.730859	ALARM:DYN004/5/ATMA/MA_AT ATMX protocol in pause : ATMX not ready	1.243
21/10/2019	10:52:16.792972	ALARM:DYN004/5/PremixMA/M/ Transfer premix WBIX in pause : ATMX not ready	1.239
21/10/2019	10:52:16.792977	ALARM:DYN004/5/PremixMA/M/ Premix in pause : ATMX not ready	1.239
21/10/2019	10:52:16.974028	ALARM:DYN004/5/ATMA/MA_AT ATMX protocol in pause : ATMX not ready	1.243
21/10/2019	10:52:26.145958	ALARM:DYN004/5/PremixMA/M/ Transfer premix WBIX in pause : ATMX not ready	4.688

Figure 3.8 – Extract of alarms for analysis of the phenomenon of overlapping alarms

Let's make a timeline as follows:

It can be seen that three alarms were triggered and overlapped, but what we are interested in is which one was triggered first and the total downtime, which is represented by the 1,419s duration in red in Figure 3.9.

- Overlapping alarms over several days

The same phenomenon as the previous one or the alarm is spread out in time, it is thus necessary to create as many days as there are between the beginning of the triggering and the end of the overlapping to have the total duration.

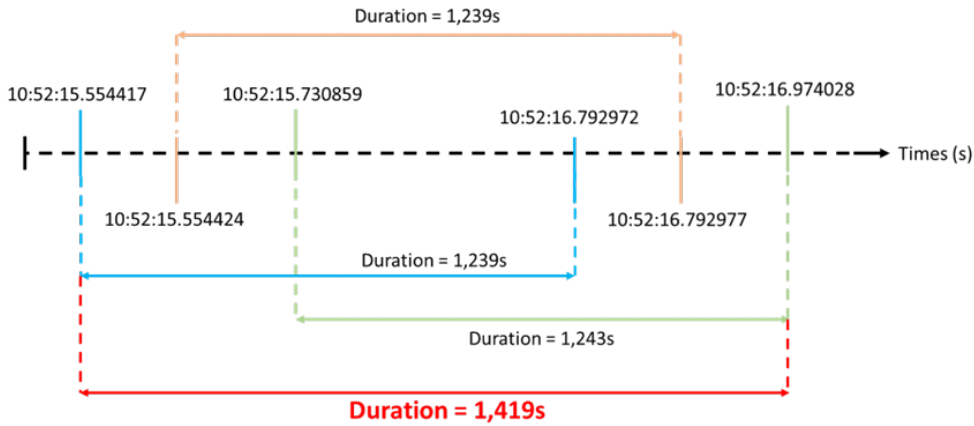


Figure 3.9 – Timeline of an overlapping alarm phenomenon

- Take into account the separation of the production lines

In our case study, there are two production lines MA and MB with common areas. It is advisable to process one set of data for each production line and a new set of data for the common part to then calculate the unavailability. Finally, the results of unavailability per zone of the production line MA with common parts in hours are obtained in Table 3.2.

Table 3.2 – Unavailability by group in hours - Dynavia

Group	Duration(h)
ATMA	62.98
CleanWater	1514.23
Diagnostic	30.53
DirtyWater	411.38
EvacuationMA	133.66
FormulatedProducts	68.83
PowderCommon	241.49
PowderMA	4.61
PremixMA	228.75
ProductionMA	1584.43
SeedsMA	58.60
Utilities	39.96
Washing	39.09

It can be seen that the two most unavailable groups are Cleanwater, which is common to both production lines, and MA production.

3.2.4.2 Pareto diagram

The Pareto principle, also known as Pareto's law or the 80-20 principle, was originated by the Italian economist Vilfredo Pareto in the 19th century, who observed that a relatively small number of people possessed the greatest amount of wealth. The purpose of this representation is to examine the data that can be divided into categories, with the aim of identifying the categories on which the user should concentrate efforts for improvement.

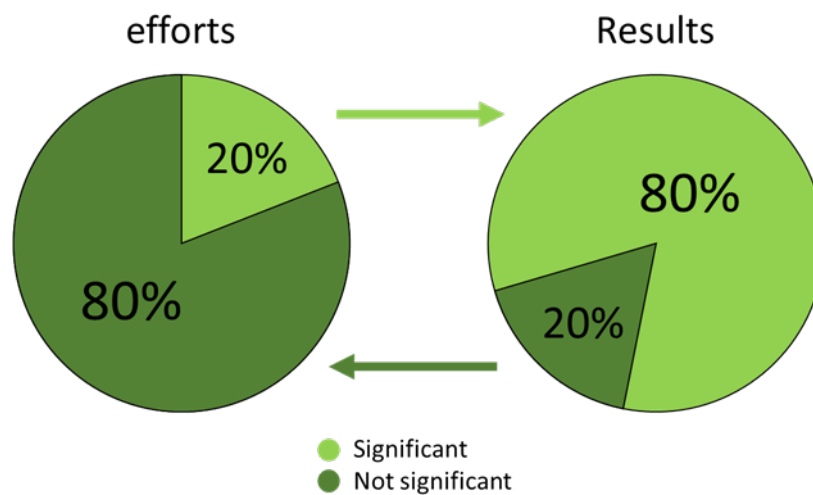


Figure 3.10 – Principe de Pareto - 80/20

Thus, this method allows us to reveal what is priority and important from what is not and therefore secondary. It is advantageous to focus on the strategic elements of a situation, such as the main causes of a problem or the elements that create the greatest value, in order to maximise efficiency and positive results.

In the context of quality improvement, Juran, who developed the quality approach based on the work of Pareto, demonstrated that 20% of the causes are responsible for 80% of the defects (figure 3.10). In other words, by focusing on the main causes, it is possible to obtain significant and lasting results in quality improvement.

The Pareto chart is a very practical, easy-to-implement graphical tool that allows a

complex problem to be broken down into several distinct components, allowing the most important ones to be clearly visualised. Note that our diagram here contains a "cumulative percentage" axis, to allow visualisation of how each component contributes to the total percentage.

When should the Pareto chart be used?

The Pareto chart is often used in the following situations:

- To understand the frequency of occurrence for a given problem.
- Assessing the relative importance of each component of the problem.
- Identify the elements that have the greatest impact on the problem.
- Determine priority areas where to focus efforts to solve the problem.

Calculate Pareto's Law

Simply enter the causes of a given phenomenon into a file and follow the steps below:

- List the major problems and their frequencies.
- Sort the frequencies from largest to smallest.
- Add a column to calculate the cumulative.
- Add a column to calculate the cumulative percentage represented by each cause.
- Select the data and plot it on a graph.

Here is an illustration with the Dynavia case study: To do this, we used our maintenance logbook, the objective is to determine which areas have the most failures.

- We retrieved the categories from the zone column and counted the number of occurrences.
- We sorted them in descending order.
- Create a new column called Cumulative and calculate the cumulative frequency of failures.
- Create a new column to calculate the cumulative percentage of each failure category.

This gives the following table:

- Plot the graph with a histogram for the frequencies of each failure (here shown in green) and the cumulative percentage via a curve (here shown in red).

Table 3.3 – Table for plotting Pareto’s law - Dynavia case

Zone	time spent (h)	Hour_Cumul
MA	36.51	33.98
FP	25.80	57.99
MB	22.13	78.59
FP2	6.86	84.98
Mx	6.71	91.23
DCS	2.00	93.09
Rapports	1.96	94.92
CI	1.31	96.15
Supervision	1.21	97.28
PO	0.83	98.06
CO	0.60	98.61
Ecran tactile	0.50	99.08
EBTM	0.41	99.47
Recettes	0.16	99.62
General	0.11	99.73
-	0.10	99.82
SiteManager	0.08	99.90
CP	0.06	99.96
Tickets	0.03	100.00

Analysis of the results

The results show that four areas represent more than 80% of the failures. The principle of Pareto’s law is to identify what the 80% on the ordinate corresponds to.

In this case, the four zones come out on top and correspond to 79.18% of the number of failures and therefore of maintenance interventions:

- The MA zone, which is the zone where happens the transformation of seeds into protected seeds.
- The MB zone, which is like the MA zone, but is another production line.
- The FP zone, the zone where the formulated and dosed products are stored.
- The MX zone, is the common problem between MA and MB.

Improvement actions will help to reduce these high failure rates. Here, the priority action for the designers and developers of the equipment of this Dynavia machine is to work on the resolution of these failures and to deepen the analyses. Ideally, this should be completed with a new Pareto chart analysis, but this time with the type of failure that

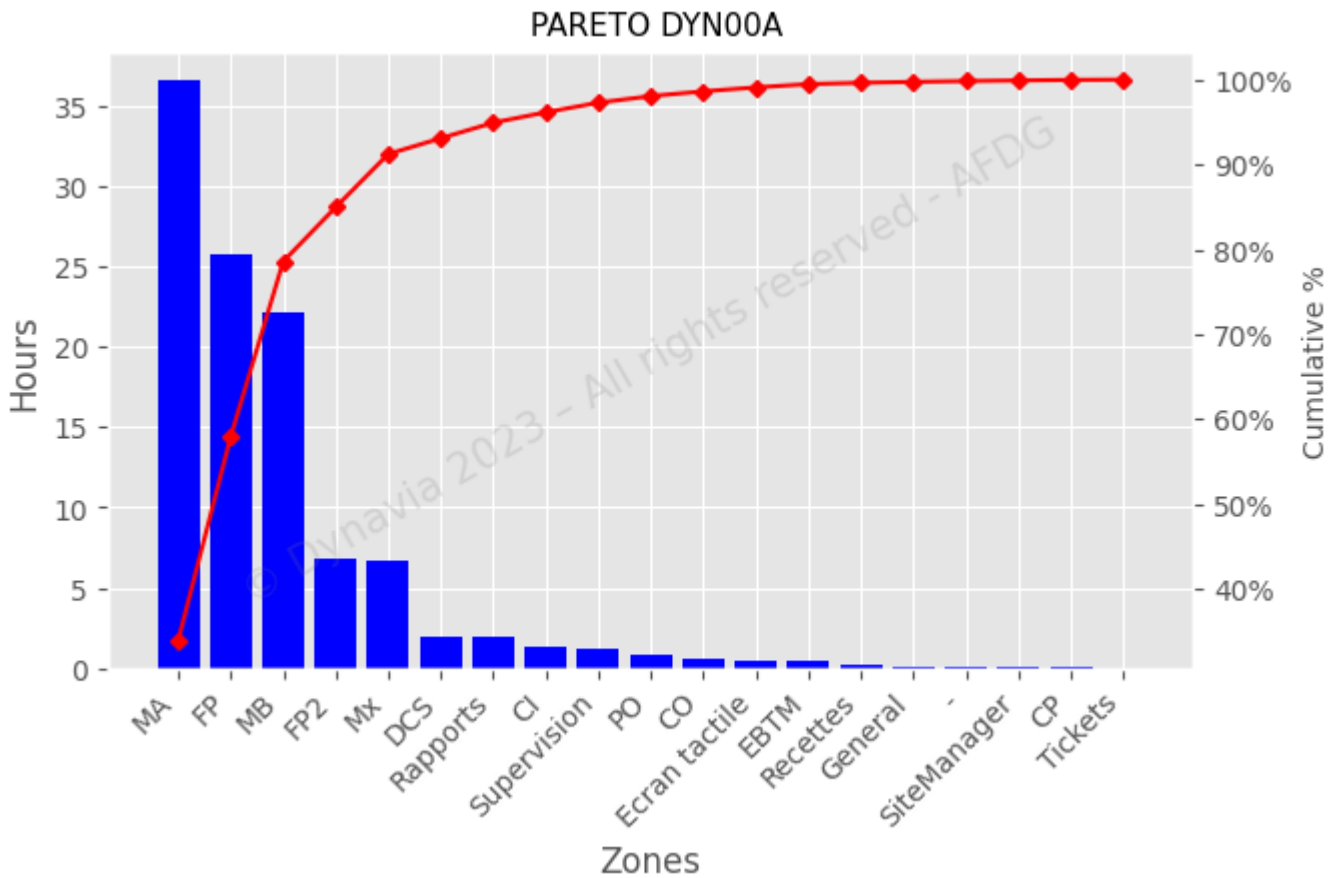


Figure 3.11 – Pareto chart DYN00A from maintenance history

has occurred. With the aim of knowing whether there are more computer, electrical or mechanical failures, etc.

3.2.5 Failure rate and maintenance history

3.2.5.1 Failure rate

In this section, we will present the reliability block diagrams (RBD) [Lasnier, 2011]. The RBD is a representation of the elements that participate in the realisation of the various functions of a system, in the form of a rectangular block, whether they are in series or parallel and linked together. It is assumed that the system works as long as the chain is not broken by the failure of the blocks.

The reliability of the whole chain is calculated according to the characteristics of the

different elements that compose it. Of course this reliability can be improved by implementing redundancies.

A system is considered redundant if it is made up of identical parallel structures, which is intended to improve operating safety. A system consisting of n elements is considered reliable if at least one of these elements is always in good working order.

There are two types of redundancy: active and passive redundancy.

- Active redundancy is when all the means are implemented together, all the elements work continuously and at the same time.
- Passive redundancy is when all resources are used on demand and the redundancy is only active when the original circuit fails.

3.2.5.2 Calculation of the system reliability

We have at our disposal two elementary structures of which we know the reliability of each component, the aim being to know the total reliability of the system.

• Structure in series

The structure is considered to be in series if one of the elements fails then the whole chain fails.

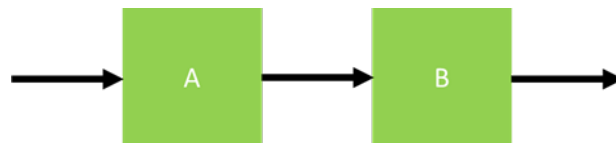


Figure 3.12 – RBD representation of elements A and B in series

Its reliability is expressed as:

$$R(t) = \prod R_i(t) \quad (3.5)$$

Its probability is expressed by:

$$P(A \wedge B) = P(A) \times P(B|A) \quad (3.6)$$

• Parallel structure

The structure is considered to be in parallel when one or more of the elements are faulty

while at least one of the elements is not. Its reliability is expressed by

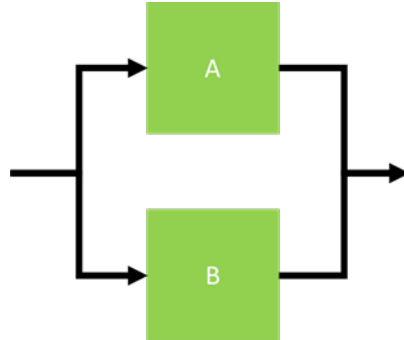


Figure 3.13 – RBD representation of elements A and B in parallel

$$R(t) = 1 - \prod (1 - R_i(t)) \tag{3.7}$$

Its probability is expressed by :

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B) \tag{3.8}$$

Now that we have all the key elements, let's move on to the representation of our system in RBD. For this, we will take the functional analysis (figure 3.5) and represent each component by blocks.

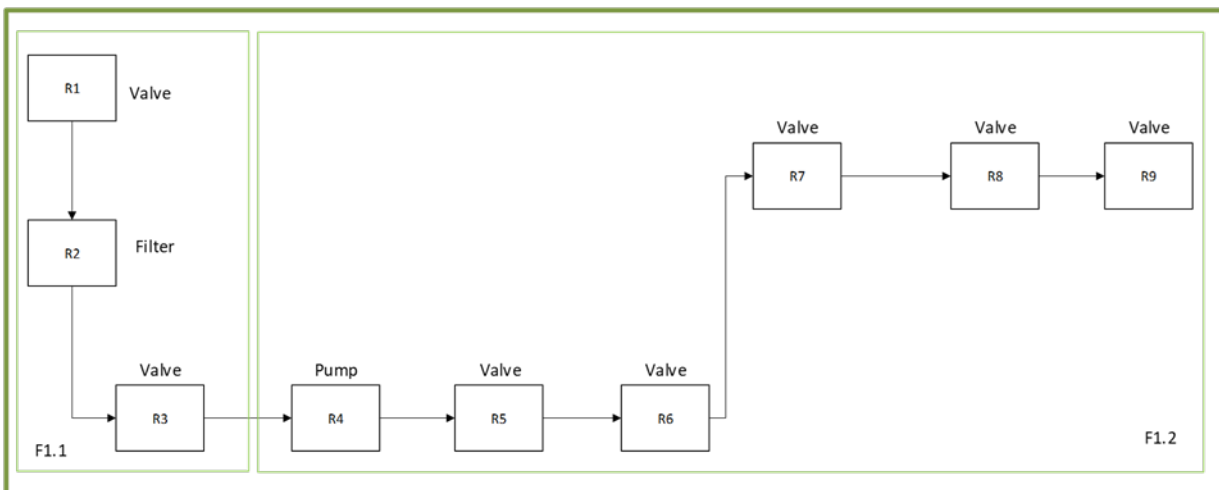


Figure 3.14 – RBD representation from the functional analysis

Our "storage formulated product" function is represented by a valve, whose reliability is named R1 that prevents the formulated product from leaving the tank or not. Followed

by a filter R2 and a valve R3 upstream of the pump.

Our "dosing formulated product" function is represented by a pump R4 followed by several valves R5 - R9 until the next function.

As we can see, all the elements are in series, there is no redundancy, if one element in the whole chain fails then the whole chain is out of order. This implies that the manufacturer, on this type of installation, must have equipment with almost perfect reliability.

If you know the reliability of each piece of equipment, because your suppliers have given you the reliability or failure rates or you have managed to calculate them yourself, it is then easy to calculate the total reliability using the RBD series structures.

If you do not have reliability or failure rates, Offshore Reability Data Handbook [Handbook, 2002] includes a reliability database with information on industrial components such as motors, pumps, valves, sensors, etc. Eight companies are behind this project, including TotalFinalElf and Shell Exploration & Production. They have formalised the representations of each component, taken readings under operating conditions and drawn up failure rate tables. Thanks to this type of chart, it is possible to have an estimate of the reliability rate of your system.

3.2.5.3 Application to our case study

We assume that the components follow the exponential reliability law and that the failure rate is constant over time, as we consider that we are in a mature period. The time base of the readings is 10^6 hours. We obtain a reliability formula noted :

$$R(t) = e^{-\lambda t} \quad (3.9)$$

For the dosing pump we selected a pump for chemical injection and a failure rate of 12.12 for all failure modes which had an operating time of 0.0961.

OREDA-2002		187					OREDA-2002				
Taxonomy no 1.3.1.1		Item Machinery Pumps Centrifugal Chemical injection									
Population 3	Installations 2	Aggregated time in service (10 ⁶ hours)					No of demands 4				
		Calendar time * 0.0964		Operational time † 0.0961							
Failure mode	No of failures	Failure rate (per 10 ⁶ hours)					Active rep.hrs	Repair (manhours)			
		Lower	Mean	Upper	SD	n/τ		Min	Mean	Max	
Degraded	1* 1†	0.24 0.23	12.04 12.12	37.31 37.67	13.19 13.35	10.38 10.41	-	26.0	26.0	26.0	
External leakage - Utility medium	1* 1†	0.24 0.23	12.04 12.12	37.31 37.67	13.19 13.35	10.38 10.41	-	26.0	26.0	26.0	
All modes	1* 1†	0.24 0.23	12.04 12.12	37.31 37.67	13.19 13.35	10.38 10.41	-	26.0	26.0	26.0	
Comments On demand probability for consequence class: Critical and failure mode: Fail to start on demand = 0.0 10 ⁹											

Figure 3.15 – Extract failure rate - OREDA injection pump - page 187 - [Handbook, 2002]

$$R(t) = e^{-\frac{12.12}{10^6} \times t} \tag{3.10}$$

For the filter we selected a filter and a mean failure rate of 0.0745 for all failure modes which had an operating time of 0.8531.

OREDA-2002		811					OREDA-2002					
Taxonomy no		Item Control system										
Population 17	Installations 13	Aggregated time in service (10 ⁶ hours)										
		Failure data		Calendar time 0.8531								
Component	No of units	Severity class				Failure rate (per 10 ⁶ hours)					Active repair time (hours)	
		#	C	D	I	U	Lower	Mean	Upper	SD	n/τ	Mean
Filter	103						0.0003	0.0745	0.2832	0.1054	-	

Figure 3.16 – Extract failure rate - OREDA filter - page 811 - [Handbook, 2002]

$$R(t) = e^{-\frac{0.0735}{10^6} \times t} \tag{3.11}$$

For ball valves we selected a valves ball and a mean failure rate of 43.79 for all failure modes which had an operating time of 9.3247.

OREDA-2002		576					OREDA-2002			
Taxonomy no 4.3.1		Item Control and Safety Equipment Valves Ball								
Population 316	Installations 18	Aggregated time in service (10 ⁶ hours)					No of demands			
		Calendar time * 9.7170			Operational time † 9.3247					
Failure mode	No of failures	Failure rate (per 10 ⁶ hours)					Active rep.hrs	Repair (manhours)		
		Lower	Mean	Upper	SD	n / τ		Min	Mean	Max
All modes	328* 328†	8.45	43.70	96.67	28.34	33.76	6.4	1.0	10.7	113.0
		8.80	43.70	100.54	29.47	35.18				
Comments										

Figure 3.17 – Extract failure rate - OREDA ball valve - page 576 - [Handbook, 2002]

$$R(t) = e^{-\frac{47.70}{10^6} \times t} \quad (3.12)$$

In the end we obtain :

$$R(t) = \prod R_i(t) \quad (3.13)$$

$$R(t) = e^{-\frac{12.12}{10^6} \times t} \times e^{-\frac{0.0735}{10^6} \times t} \times (e^{-\frac{47.70}{10^6} \times t})^7 \quad (3.14)$$

Let's plot the reliability curves of each component and the total reliability over time to observe the evolution every 100 hours (figure 3.18).

Of the three reliability functions, the valve function decreases faster than the pump and filter functions. It is therefore potentially the most unreliable of the three. The reliability of the filter is almost perfect. For a functionality of 100 hours the pump, filter and valve have a reliability of 99.87%, 99.99% and 99.56% respectively. This gives a reliability of 96.86% for the whole chain. After 1000 hours, the reliability of the components is still high at 98.79%; 99.99%; 95.72% but the total reliability has dropped to 72.75%. The total reliability curve falls rapidly in the first 5000 hours, the manufacturer has every interest in maintaining his production line, applying preventive maintenance in the first instance to monitor all the equipment and particularly the valves.

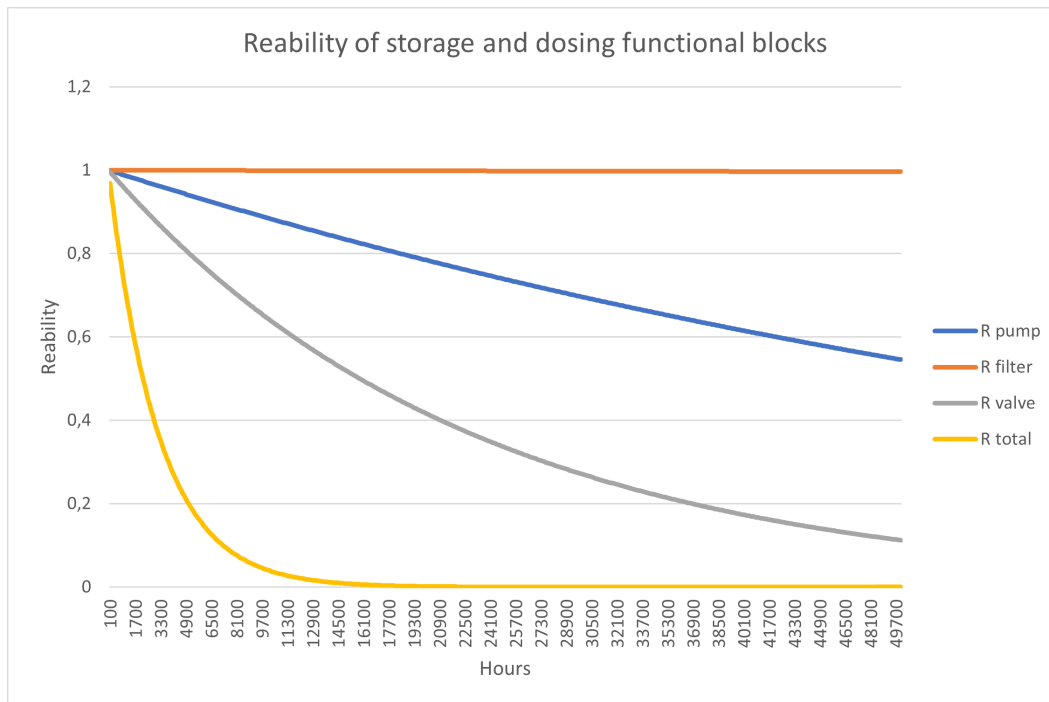


Figure 3.18 – Block diagram of the reliability of storage and dosing of formulated products from the BDF representation 3.14.

3.2.5.4 Maintenance history

An indicator that can also help to determine the candidate for predictive maintenance is the search of the maintenance history (see section 3.2.4.1). This is because it allows us to know if the same equipment has failed several times using the TAG column and how long it has been in service each time.

In this way, we have access to the type of failure that occurred and the details of the intervention, which will help in the selection of data. In the case of Dynavia, the system being reliable, the only component that has failed the most is a pump, which has only failed three times. It is therefore difficult to practice predictive maintenance with so few cases. Nevertheless, the system will age over time, and it is likely that new failures will occur on it or in different places, but Dynavia will be able to cope with them.

3.3 Data collection and preparation

3.3.1 Mapping the data sources

Now that the administrative steps are over, you will be in contact with the IT department for further steps, but this time in a phase of recovering the raw material for your study.

In most large companies, you will never have direct access to the machine's data for reasons of cyber security and to avoid disrupting ongoing production. The IT department will be able to show you the data sources, but also tell you what they contain such as:

- A database containing product information, manufacturing recipes, etc.
- A chronotrend database which records all the information of the sensors, actions, processes of your system. In the form of the identifier, date, time of recording and value. The frequency of recording can be in different ways and this will obviously have an impact on the predictive maintenance work. The data can be recorded by the triggering of an event, by a percentage of the evolution of the last value or by a fixed frequency.
- A chronolog database that records all alarms triggered when there has been any failure. These are often presented in the form of priority, start, end, trigger zone, description.
- The audittrail, which makes it possible to know any action that has been taken on the supervision by a production or maintenance operator. This data is generally in the form of a table which indicates the zone of the supervision, the date and time, the old value, the new value written by the user, the name of the user. This data can be interesting to determine the entered production order, to determine the production start, the product change...etc

3.3.2 Data extraction

The maintenance file allows us to have the minimum of information to extract from the different databases the information we need to create a maintenance data history. Of course, depending on the architecture of your data, it is important to list them in relation to your functional breakdown of the system in order to maintain consistency.

Using the time index of the start of the intervention, where the failure is, and the end

of the intervention, it is possible to extract the data, by creating a time filter with the programming language of your choice, analyse it and keep only the significant data.

This process can be repeated until a sufficient quantity is obtained to create a dataset to train a model on well-defined cases.

3.3.3 Preparation

The IT department has given you access to the data and their descriptive documents. From the outset, put in place good practices with the implementation of automatic data extraction procedures according to your needs:

- Export the data source to your workspace in a working folder dedicated solely to a specific case study. If you are working on another case study then a new folder architecture must be set up. The data in the source folder only needs to be copied. In this way you avoid working directly on this data and changing it and losing the original data, because the IT department will not be able to make extractions for you all the time.
- Analyse the structure of each data source.
- Create programs that allow you to adapt the format of the data into a format that you can use.

In the case of Dynavia, the data structure is not the same for sensor/actuator data, alarms and audit trail data. XML parsers had to be created to CSV format.

3.3.4 Modeling

This section consists of setting up all the necessary steps to create the predictive model. We have used and explained this method in chapter 4 concerning the experimentation on an industrial dosing pump in the section 4.3. To achieve this, the data must be processed, adapted to the necessary format for the chosen machine learning algorithm with a training and test data set. Then comes the final training step so that the model can represent the behaviour of the data. During the first stage of creating the model, it is necessary to compete with different algorithms depending on the type of prediction you want to make, in order to determine which one scores best. For example, if you want to predict the remaining life of a system or when it is likely to fail, you would use the regression method. On the other hand, if you want to predict a category, such as whether the current data is a failure or not, or what type of failure may occur, you would use classification.

3.3.5 Evaluate

3.3.5.1 Setting up a machine learning process

In the first phase of model creation, all the algorithms, depending on the type of prediction desired, should be put into competition to see which one has the best score. Then the winner will be the model chosen for the new failure predictions.

3.3.5.2 Maintainability process of the machine learning model

Now that the data is loaded, cleaned and the model is created and trained for the first time, it is time to evaluate it per period with a test dataset (figure 3.19).

Whether you are in a case of prediction following a classification or regression training, the methodology remains the same. A threshold of acceptability for each evaluation method must be defined to ensure monitoring. Let us illustrate with an example of classification and the associated metrics, see section 2.2.5 (non-exhaustive list):

- The model tests new input data, the value of the score varies, but it remains above the acceptability threshold, so the model is still functional and remains in this "evaluation" phase and repeats this process as long as it does not cross the threshold.
- If the variation in the score is below the threshold of acceptability this time, then the model is no longer reliable. Before re-training the model, it is a good idea to check whether the data that has been tested is noisy or of poor quality, which results in poor prediction and a poor score. In this case, it would be wise to retest with a new set of data if this persists then has the model been poorly trained? The model is too complex or not complex enough which affects its ability to produce reliable predictions. It is time to select a new dataset to retrain, test it with a real dataset and analyse the new score, if it is back above the acceptability threshold then the re-training is complete.
- Last case, the score at the evaluation stage is not good, the training of the model with a new dataset has not been successful. The whole machine learning process has to be repeated from the beginning with a new selection of training data, rebuilding the model using a different architecture or even changing the machine learning algorithm, training it again, testing it and evaluating the new predictions and then putting it into operation.

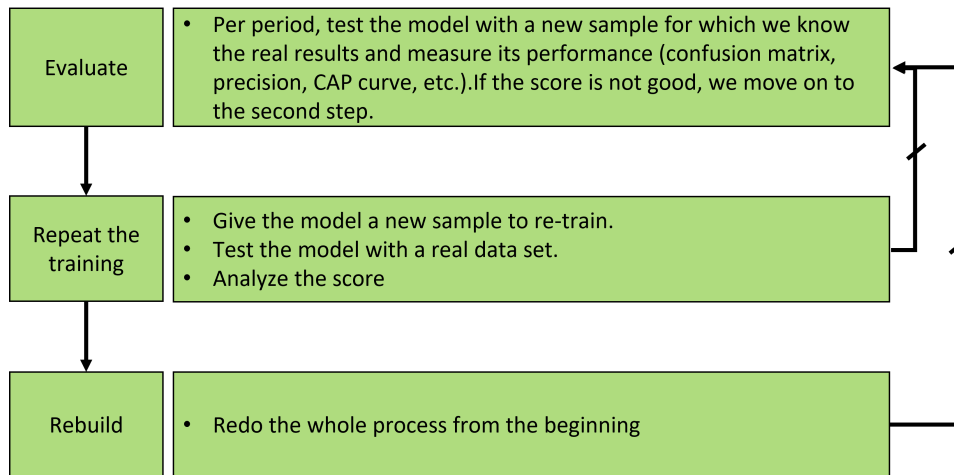


Figure 3.19 – Maintainability process of the machine learning model

3.3.6 Maintenance decisions and actions

3.3.6.1 Presentation of prediction results and KPIs

The prediction of possible failures by the model poses a new problem: should we trust this prediction and put in place actions and maintenance means when we are not sure that it will happen? This trust is often difficult to establish at the beginning due to the lack of experience and feedback on the use of this technique, but with time and production issues.

If the system warns you of an impending failure of the associated equipment, please monitor and analyse the warning signs such as increasing pressure peaks, a continuous decrease in efficiency and the appearance of increasingly audible noise...etc that could put you in confidence on the model and put in place maintenance well before failure but also at the time when there is no production, with verification of spare parts in shops, human resources to face it. It is true that you are taking a financial risk by changing equipment that may not break down, but you will always lose less than if, in addition to that, you lose the loss of production earnings, which can be measured quickly in kilos of euros/hour.

3.3.6.2 Reflection

There can be two dangerous cases of prediction:

- In a classification case, the model predicts that there will be no failure based on the input data, yet there is a real failure. The dangerousness depends on the domain and

the circumstances in which this technique is used. Because if we take as an example the nuclear field and the model predicts that there is no failure when there is a real radioactive leak, the dangerousness is immediately noticeable. The maintenance team will not be prepared and will then find itself in a corrective maintenance situation, having to act as quickly as possible to get the system back into service.

- Another case, similar, but using regression, the model predicts an end of life, a failure date, but it occurs earlier than the prediction.

In all cases, dangerous or not, it is better to remain vigilant about the predictions of the machine learning model even if the confidence in the predictions is high. It is preferable to always monitor the industrial park, to follow the evolution of the sensitive process data.

In summary, predictive maintenance is not ultimately intended to replace all other types of maintenance, because its strength and also its weakness, its predictions remain only eventualities that may or may not occur. It is rather intended to complement preventive maintenance and to avoid corrective maintenance as much as possible.

3.3.6.3 Back to the design

Despite the maintenance actions on the part of the system concerned, the model predicts that there is still a risk of failure. Either the model is right or it is wrong, this can be validated by experience. However, if the problem persists, then that part of the system has been poorly designed or the customer is not using it under the conditions defined by the manufacturer. It is then necessary to go back to the design phase to rethink the part of the system to possibly change the technology, oversize it, to respect the conditions of use in order to meet the function.

3.4 Conclusion

In this chapter, we have proposed a predictive maintenance strategy applied in an industrial case. We have presented a workflow of our method starting with the stage of understanding the study and situating it. For this, we proposed a set of analysis methods with a funnel objective, i.e. starting with global analyses and reducing the scope until identifying a work area on the system and finally identifying a potential candidate to

apply predictive maintenance. Because the implementation of predictive maintenance has a financial and time cost. This is followed by the data collection and preparation stage necessary for the modelling stage. Once the predictive model is created, it must be evaluated and maintained to ensure that the predictions are always correct. These predictions only make sense if the results are analysed and maintenance decisions are made.

ANOMALY DETECTION BASED ON MACHINE LEARNING

4.1 Introduction

The integration of elements of the smart factory (such as autonomy, augmented reality, Internet of Things, Big data and artificial intelligence) in all stages of industrial processes is starting to have more and more importance [Julien and Martin, 2021].

Predictive maintenance is a set of techniques designed to help determine the condition of in-service equipment in order to estimate when maintenance should be performed on it. It intervenes the most upstream of the failure and is therefore the most effective as it allows for an increased availability of the equipment coupled with a reduced financial impact [Gu et al., 2017].

In this context large volumes of data are collected, potentially through an Internet of Things (IoT) type of interface. This data can come from sensors, actuators, alarms, process steps and can be linked to a machine learning model in order to make predictions of the remaining useful life-time (RUL) of a specific equipment [Li et al., 2018b], [Zhang et al., 2019a] or it can also be used to detect anomalies like failures. The purpose of anomaly detection is to identify anomalous patterns in the data related to an industrial process. Anomaly patterns are difficult to identify because of the volume of data and the temporal relationships that may exist between patterns. For these reasons, traditional techniques often fail to identify anomalies.

In recent years, anomaly detection using deep learning algorithms has become increasingly popular and was met with some success thanks to the use of Long Short-Term Memory (LSTM) [Cheng et al., 2021] and Gated Recurrent Unit [Qu et al., 2018]. The

LSTM Autoencoder for multivariate time series is used for the reconstruction of the characteristics of a data set [Nam et al., 2020].

In this chapter, we propose an anomaly detection method based on machine learning models to find anomalies and compare the results with three real anomalies in the case of an industrial dosing pump. The remainder of this chapter is divided as follows. Section 4.2 illustrates the industrial case and highlights some of the anomalies we are trying to detect. Section 4.3 presents the deep learning methodology we used. Section 4.4 shows the implemented experimentation and finally, in Section 4.6, we conclude with several remarks.

4.2 Industrial Use-case

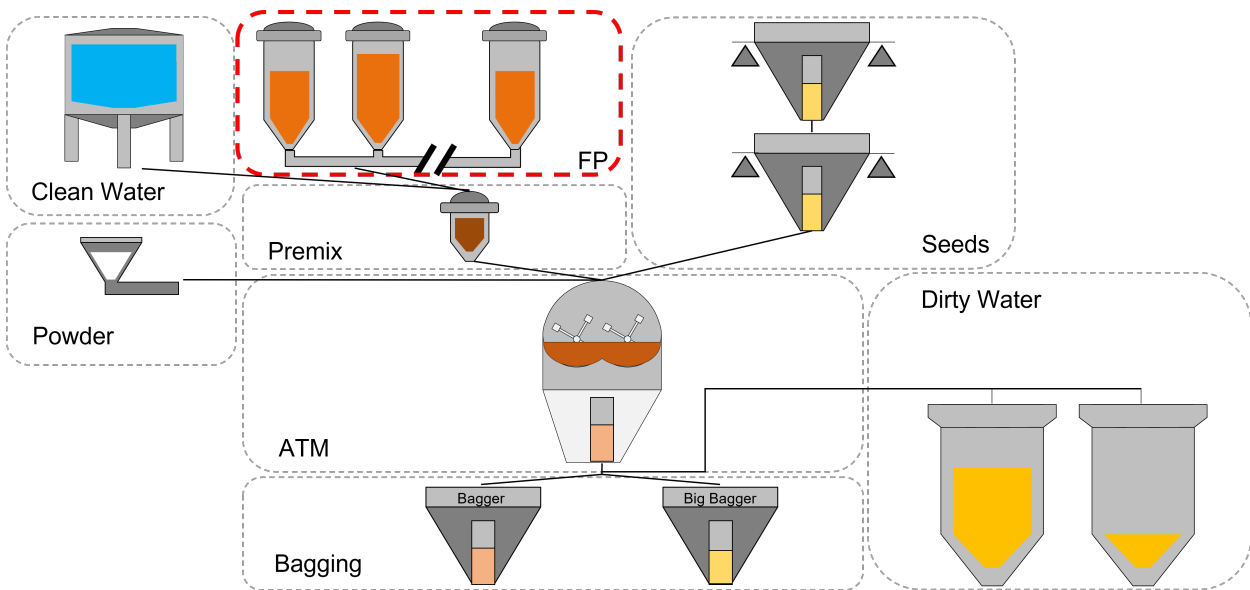


Figure 4.1 – Process Dynavia

In order to understand the case study, Figure 4.1 gives us an overview of the Dynavia process and situates the study in relation to the rest of the installation. The following is a description of each area:

- Seeds: allows the dosing of seeds for each treatment. The upper hopper stores the seeds for the first time and then pours them into the lower hopper, where we will accurately dose the next treatment.

- Formulated Products: Here we have the storage of the formulated products in tanks. During a treatment, the dosing pumps will take the necessary quantity of product corresponding to the quantity of seeds.
- Clean water: used to store hot water and distribute it throughout the installation, which is mainly used for cleaning.
- Powder: allows the dosing of powder for each seed treatment.
- Premix: the formulated products are then incorporated into a tank, mixed and pressurised to be injected into the ATM.
- ATM: the seeds fall into the ATM and are agitated. Nozzles inject the mixture of formulated products and powders onto the seed during the process and will eventually be dried with dehydrated air.
- Dirty water: This area allows the recovery of waste water from an insecticide or fungicide treatment during one of the Dynavia's washes. The water is then evacuated for destruction or it is reused for the same recipe.
- Bagging: the seed treatment is now complete, the ATM hatches open, the seeds will be bagged in bags or be in a large bin to be sold.

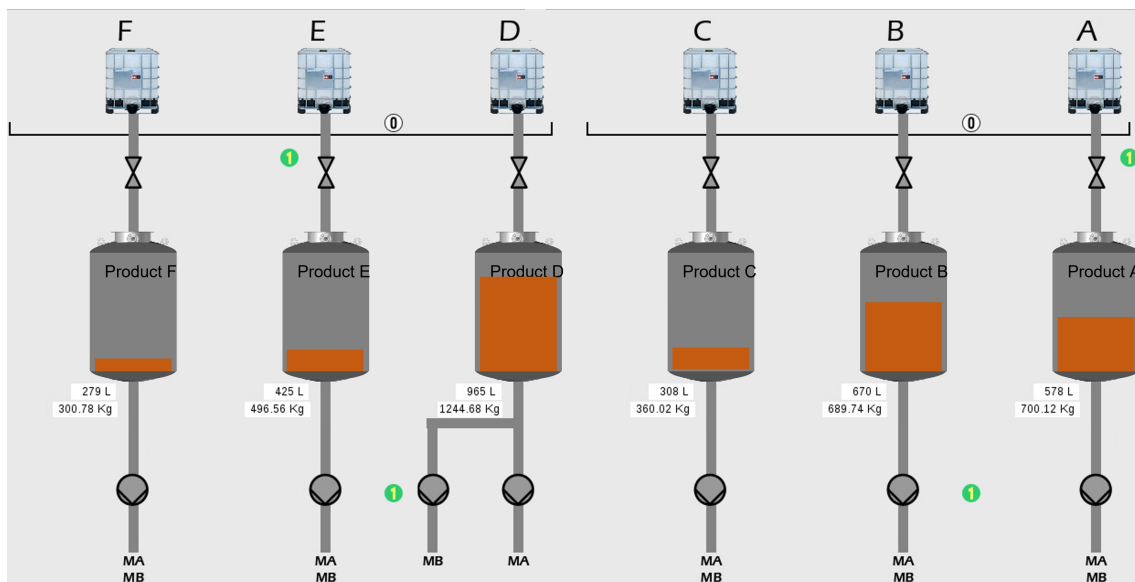


Figure 4.2 – An extract from the FP rack

The FP area circled in red (Figure 4.1) is composed of a set of 15 macro-racks for large quantity dosages often used and a micro rack for more specific productions requiring a

small quantity of product which can be used for two production machines MA and MB. Figure 4.2 shows an extract where each line, named with the letters of the alphabet, consists of an IBC (Intermediate Bulk Container) containing the formulated product which is then poured into a stainless steel tank for storage and future dosing without risk to the user. The number of lines is important because it allows formulated products to be loaded at the same time and therefore different recipes to be produced one after the other without having to change products during production or between productions, which would be a waste of time, human resources and money. Thus a product that is used most often will remain in the same tank, on the same line.

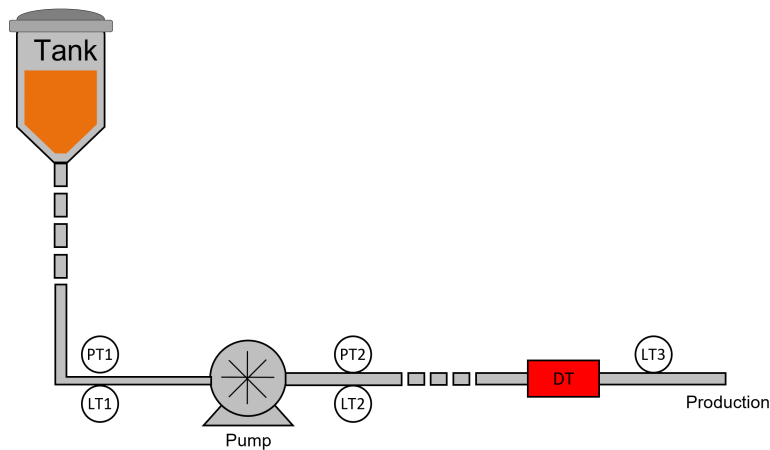


Figure 4.3 – Piping and Instrumentation Diagram (PID) simplified - Dosing pump

To understand the case study, Figure 4.4 gives a schematic view of a simplified version of the PID of a single line dosing pump. We observe a tank, holding a particular substance, that is connected to a pump leading further on to the production area. Two pairs of level (LTr) and pressure (PTr) sensors may be found both before and after the pump, while a density meter (DTr) and a third level sensor may be found further down. Table 4.1 summarizes these sensors, the type of data they provide as well as the number of data observations across a sixteen month period.

The different usage scenarios

Each line in the FP area can be used in the following cases

- The loading of the line with formulated product. The product stored in the tank flows by gravity until it reaches the pump. It is then detected by LT1 while PT1

Table 4.1 – Available data

Captor/Actuator	Data	Units	Nb records
LT1	Level	Boolean	480
LT2	Level	Boolean	486
LT3	Level	Boolean	20458
PT1	Pressure	Bar	2751197
PT2	Pressure	Bar	3109809
DT	Density	Kg/m ³	4227
Pump	Speed	rpm	1090629

notices a slight increase in pressure. When the pump is turned on, the product flows downstream of the pump and is detected by LT2 and PT2, while DT gives the density of the product. LT3 will confirm the presence of product at the end of the line so that the dosing process can begin.

- The unloading of the line in formulated product, the principle of which is to empty the whole line thanks to the pump which starts in reverse, so the product circulates in the opposite direction. LT3 and LT2 do not detect the presence of the product and go to the state 0 and the pump stops. LT1 remains at 1, because there is no more product downstream of the pump and it must not run empty.
- The dosing is executed when a production is started, the line is loaded with product up to LT3. The dosing pump starts up and injects the required amount of product for each batch of the current recipe.
- Cleaning, as we mentioned at the beginning, we have simplified the PID, normally valves are present to allow the cleaning of each part of the line and to evacuate the waste water. This step is crucial, because if there is a change of product on the same line, the pipes and components must be clean, without any trace of another product formulated to avoid any contamination.

Maintenance

When the production operator configures the equipment with the will to launch a production or during a production, if he encounters a failure that prevents the good progress, the system pauses and launches an alarm of the supervision. If the production operator cannot solve the problem with the maintenance operator, there are several possibilities depending on the criticality and urgency:

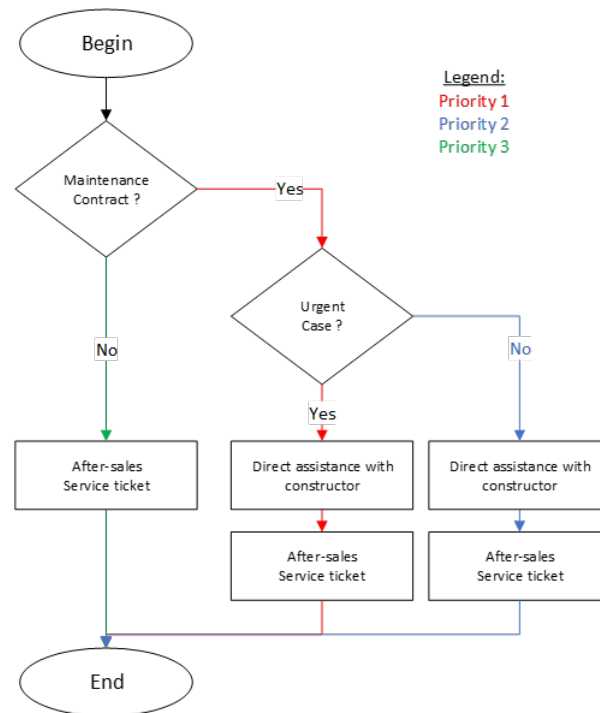


Figure 4.4 – Piping and Instrumentation Diagram (PID) simplified - Dosing pump

- If the customer is under a maintenance contract and the situation is urgent (Priority 1), then he calls the manufacturer’s after-sales service phone line to provide assistance and a maintenance ticket is opened until the problem is resolved.
- If the customer is also under a maintenance contract, but the situation is not urgent (Priority 2), then he can call the manufacturer’s after-sales service phone line and he can open a maintenance ticket but it will not have priority if there is a priority 1 case.
- If the customer is not under a maintenance contract (Priority 3), then he does not have priority over the first one because he can only open a maintenance ticket.

As a manufacturer, we have an online maintenance ticket platform available for each customer. So we analyzed them to find cases of failure on the same equipment for the same customer because from one customer to another, the system is not necessarily used with the same raw materials, the same products and the customers may not have the same machine, because they have specific needs. Therefore, the customer DYN00A is chosen for this study, because we received three maintenance tickets (table 4.2) on the

same dosing pump with the same formulated product over a period of 482 days.

Table 4.2 – Maintenance tickets received

Date	Time	Description
2020-09-08	1:13pm	Line loading problem
2020-11-19	12:39pm	Dosage problem
2020-12-03	4:45pm	Overpressure and leak detection

The aim of this study is to find the warning signs of these failures within the collected data in order to avoid failures in the future. We begin by manually analyzing two maintenance cases on the same pump.

4.2.1 Case study: Line loading problem

On 2020-09-08, there was a maintenance order regarding a loading issue on the production line. Figure 4.5 illustrates a correct case of line loading while Figure 4.6 illustrates the behavior of line loading when an anomaly occurs. Both figures depict the product density, as given by the density meter in red, and the pump speed in blue. The density axis (Kg/m^3) can be found on the left hand side, while the pump speed axis (rotations per minute) is on the right hand side.

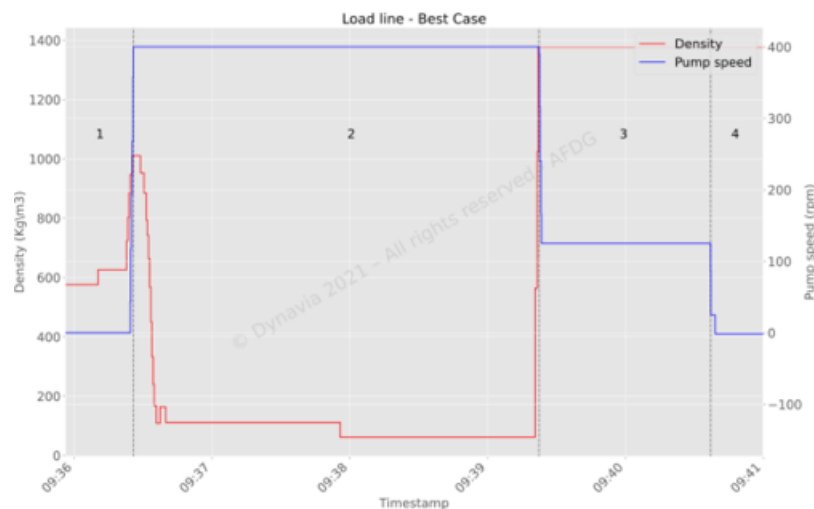


Figure 4.5 – Case 1 - Correct Line loading

In the first instance we have the correct case where:

1. The line loading process begins and the pump turns on, having initially a rotation speed of 0 rpm.
2. The pump speed reaches its maximum 400 rpm and the density of the product decreases to almost zero; the product flows for around three minutes.
3. The pump starts to reduce its speed to 150 rpm as it has almost finished loading the line; the density increases again to 1400 Kg/m³.
4. The pump stops, reaching a rotation speed of 0; the density is at its maximum and the line is fully charged with product.

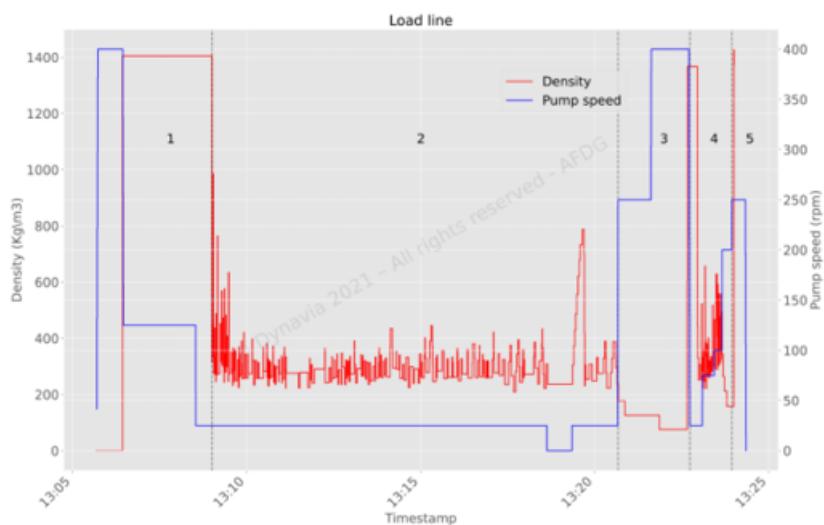


Figure 4.6 – Case 1 - Line loading problem

Then in a second instance we have the explanation of the line loading problem in Figure 4.6:

1. At the beginning the process runs normally, the pump slows down when the density meter detects the right density.
2. The density is no longer detected by the density meter as if the product was moving backwards.
3. The pump speed is forced by the technician at 400 rpm to add more power to load the line but usually the standard speed is sufficient.
4. Forcing is stopped once the density is detected.
5. The speed is again forced to 250 rpm to finish loading the line, even if this means losing a small amount of product to the next equipment due to inertia.

4.2.2 Case study: Overpressure problem and leak detection

In Figure 4.7, the graph describes the behavior of pressure in a process over time and the occurrence of a leak at the pump outlet. The graph on the left shows the normal pressure behaviors following the process, with pressure peaks becoming more intense as time passes. On December 3, 2020, at 4:45 pm, a leak occurs at the pump outlet, which is indicated by the beginning of the red zone on the graph. This event leads to the replacement of the pump on December 8, 2020, indicated by the end of the red zone. After the replacement, the pressure patterns return to normal. It's important to note that the alarm threshold for extreme pressure, with a priority of 134, is calibrated at 16 bars.

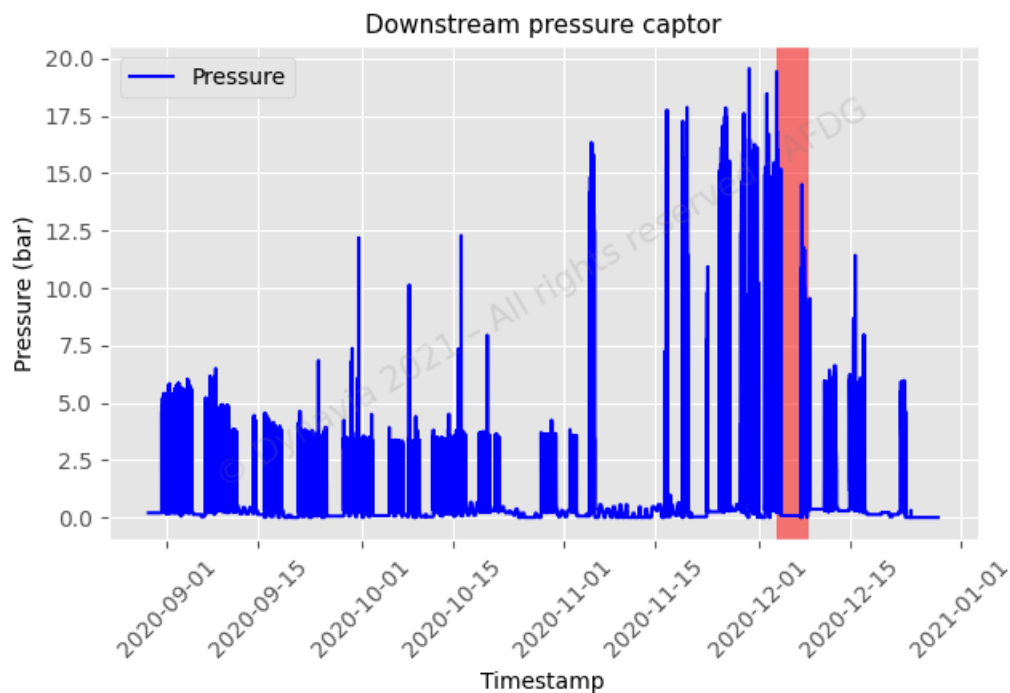


Figure 4.7 – Case 2 - Evolution of overpressures and leaks

4.3 Methodology

To detect several anomalies of different types automatically from the data, we use machine learning techniques. Figure 4.8 illustrates the methodology for detecting anomalies with the acquisition, processing and standardization of data followed by the creation of the model with its evaluation in the training and test phases in order to evaluate its performance.

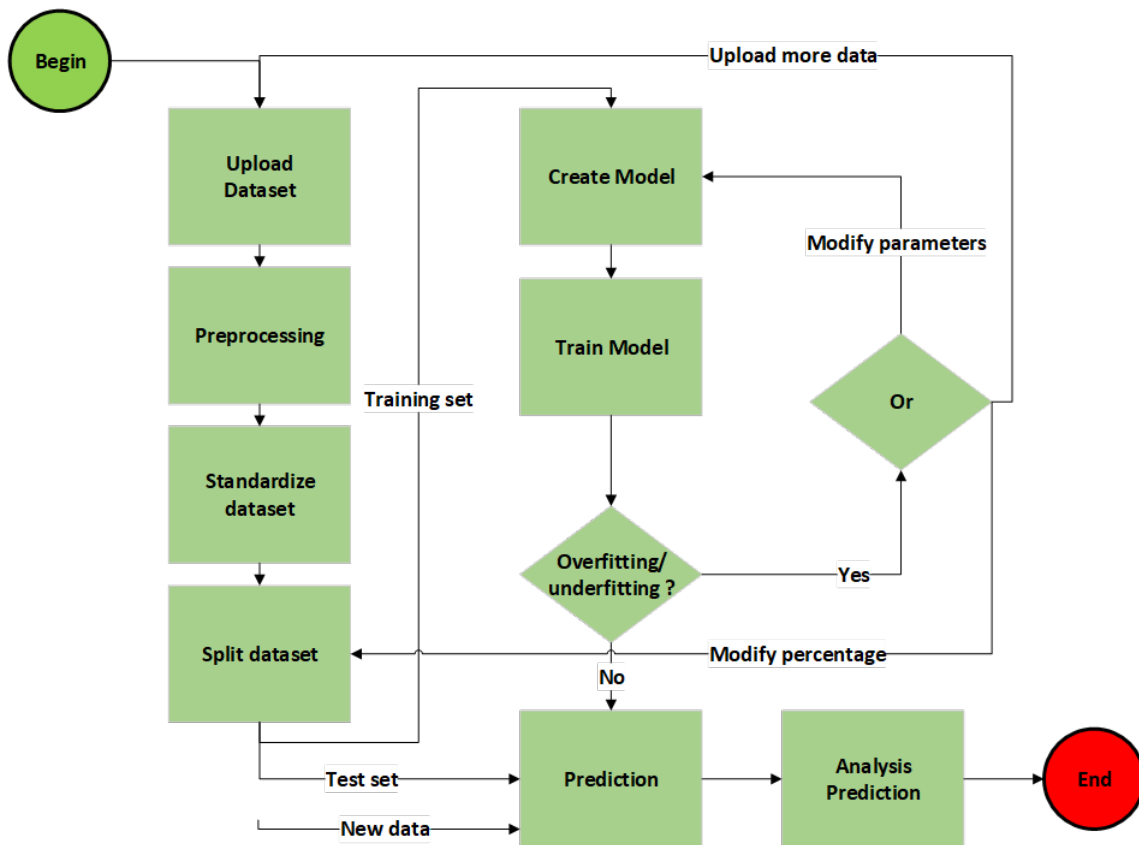


Figure 4.8 – Process machine learning

1. Upload dataset: this step allows to select the significant data and its quantity required for our case study among all equipment.
2. Preprocessing: once the data is loaded, this part consists of successive treatments like cleaning extreme and missing values, data fusion and resampling.
3. Standardization: in our dataset we have different kinds of data with different scales. Since LSTMs, and Deep Learning methods in general, require data to be standard-

ized, which involves rescaling the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1.

4. Splitting the dataset into a training set and a test set: if we have only one dataset, it is necessary to cut it into two parts, one for training (about two thirds) and one for testing (the last third).
5. Model creation and training: it is necessary to create several deep learning algorithm models and with different associated characteristics in order to put them in competition during the training phase and choose the one that has the best performance.
6. Prediction/Anomaly detection: once the model has been trained with the best parameters, it remains to make the detection of anomalies on all the test data and to compare the results with our real cases.

4.3.1 Upload dataset

This step allows to select, among all the equipments, the significant data and their quantity required for our case study. In DYN00A, the data corresponding to the density does not have the same name as the one on the PID. In order to identify it, we have to use a supervision tool called "Trendviewer" which allows us to visualize the curves of each equipment or some data of the industrial process, see Figure 4.9. The tool provides us

Groupe/Point de mesure	Description	Système Runtime	Tag n°	Unité
FormulatedProducts				
MB_LineE_DosageLineE_DT_01_Density_Val	Desc	run (run)	MB1_DT_01	kg/m3
MB_LineE_DosageLineE_DT_01_Density_ValRaw	Desc	run (run)	MB1_DT_01	kg/m3
MB_LineE_DosageLineE_DT_01_Density_ValRawUnScal	Desc	run (run)	MB1_DT_01	kg/m3

Figure 4.9 – Trendviewer - Data identification

with the filter where we enter the process reference of the pressure sensor (1), then we observe the different variables associated with it. In our case, it is its analog value (2).

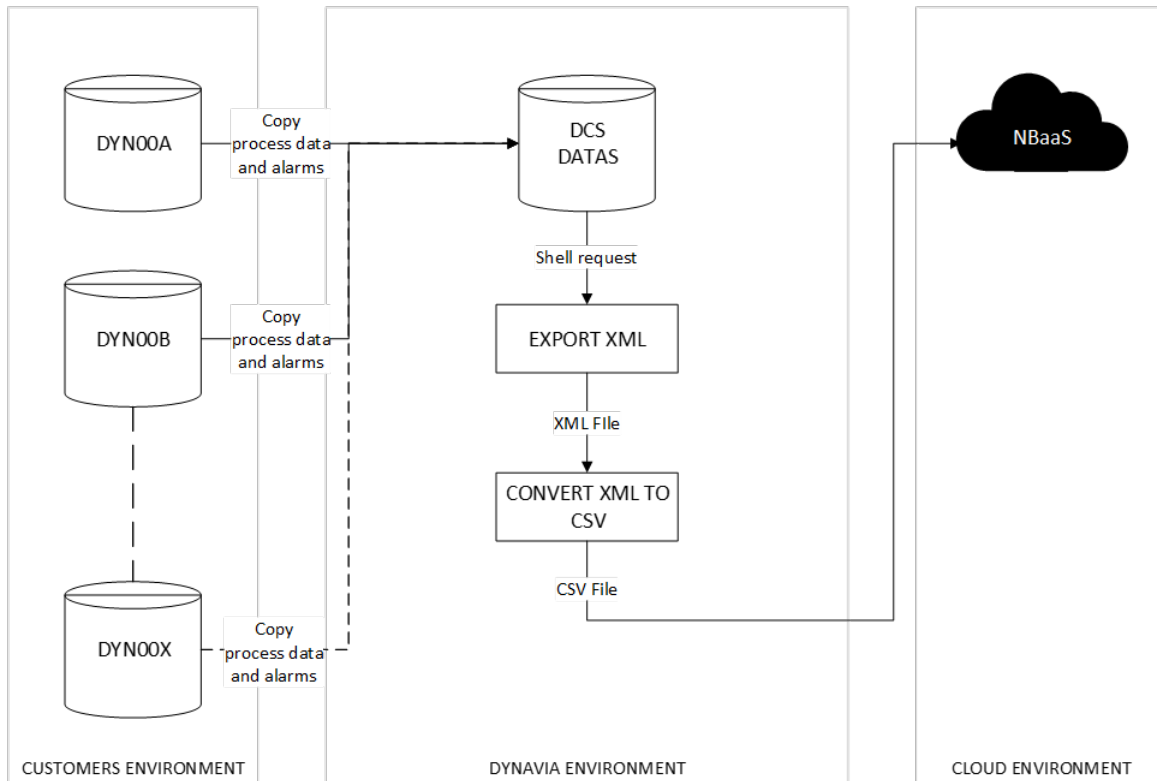


Figure 4.10 – Transfer of customer data to a NBaaS cloud solution

Now that we know the name of the data we are interested in, we can export it in Extensible Markup Language (XML) format, imposed by the APROL process control software of each Dynavia machine, via a shell request from the DCS-DATA. We cannot use the XML file in this state for our study. Therefore, we developed a Python code `ConvertXMLtoCSV` to convert the XML file (Figure 4.11) to CSV format (Figure 4.12) for use with the python library Pandas (mentioned in the section 2.2.1). Then, we transfer the CSV file to a cloud computing solution NBaaS (mentioned in section 2.2.2) from OVH. This whole process is represented in the Figure 4.10.

```
<record id="TRD:MB_LineE_DosageLineE_DT_01_Density_Val">
  <ts date="2019-09-03" time="07:10:52.883196">348.836364746094</ts>
  <ts date="2019-09-03" time="07:12:09.369981">348.423583984375</ts>
  <ts date="2019-09-03" time="07:12:22.023785">348.670471191406</ts>
  <ts date="2019-09-03" time="07:32:52.643708">348.059967041016</ts>
  <ts date="2019-09-03" time="07:33:23.079720">348.114105224609</ts>
```

Figure 4.11 – Example Dataset - Density in XML format

This program is generic, because all process data of sensors, actuators, process steps

```

id,date,time,value
TRD:MB_LineE_DosageLineE_DT_01_Density_Val,2019-09-03,07:10:52.883196,348.836364746094
TRD:MB_LineE_DosageLineE_DT_01_Density_Val,2019-09-03,07:12:09.369981,348.423583984375
TRD:MB_LineE_DosageLineE_DT_01_Density_Val,2019-09-03,07:12:22.023785,348.670471191406
TRD:MB_LineE_DosageLineE_DT_01_Density_Val,2019-09-03,07:32:52.643708,348.059967041016
TRD:MB_LineE_DosageLineE_DT_01_Density_Val,2019-09-03,07:33:23.079720,348.114105224609

```

Figure 4.12 – Example Dataset - Density in CSV format

have the same XML structure. So to export other data, we just need to change the name of the input and output variable. And if we need to convert a set of XML files to CSV at once, we have another Python program available.

The loading of the data is done via the pandas library with the help of the "read_csv" function which will put the data in spreadsheet form (DataFrame). Thus Figure 4.12 is in two parts, the first part concerns the first line, made up of "id, date, time, value" which are the headers of the columns of our DataFrame. The second part consists of the data associated with the columns, we can see a representation of the dataframe in Figure 4.13.

	id	date	time	value
0	TRD:MB_LineE_DosageLineE_DT_01_Density_Val	2019-09-03	07:10:52.883196	348.836365
1	TRD:MB_LineE_DosageLineE_DT_01_Density_Val	2019-09-03	07:12:09.369981	348.423584
2	TRD:MB_LineE_DosageLineE_DT_01_Density_Val	2019-09-03	07:12:22.023785	348.670471
3	TRD:MB_LineE_DosageLineE_DT_01_Density_Val	2019-09-03	07:32:52.643708	348.059967
4	TRD:MB_LineE_DosageLineE_DT_01_Density_Val	2019-09-03	07:33:23.079720	348.114105

Figure 4.13 – Example Dataframe - Density

4.3.2 Preprocessing

With the dataset loaded, we will expose the pre-processing that we have applied on the input data, the measured density of the formulated product.

- The merging of the columns "date" and "time" to obtain a new column "timestamp". The format of the date is: 'YYYY-MM-DD' (YYYYY: Year; MM: Month; DD: Day). The time format is: 'HH:MM:SS' (HH: Hour; MM: Minute; SS: Second). We obtain the following format for the timestamp: 'YYYY-MM-DD HH:MM:DD'.

- Delete the "id" column, as it does not provide any relevant information for our study since we have a data and we know which one. This allows us to save storage memory.
- Resample the signal entirely, because Dynavia records the information from the sensors/actuators not by a given sampling frequency, but by a percentage of variation of the physical value (here about 0.1%). We used a sampling rate of every 30 seconds for this study. It is crucial to consider that using too high a sampling rate may result in a loss of important information for the study, especially with regard to the three failures. Conversely, a sampling rate that is too low will result in a significant increase in the volume of data, which will also increase the computation time and require more storage resources and computing power.
- Another treatment consists in decomposing the "timestamp" variable into several temporal characteristics such as the hour of the day (f2), daylight (f3), the day of the week (f4), working week (f5). The purpose of this technique, also known as "date encoding", is to capture complex temporal patterns, represent temporal relationships and manage temporal cycles, which can be essential for good performance.

We obtain a new dataset of dimensions :

Table 4.3 – Comparison of dataset dimensions before and after preprocessing

Old	New (after preprocessing)
(4227 rows, 4 columns)	(1389031 rows × 5 columns)
[<i>id, date, time, value</i>]	[<i>value, hour, daylight, dayWeek, workingDay</i>]

4.3.3 Standardization

Our dataset consists of a column labeled 'value' with values in the order of thousands, while the values in the other columns are in the order of units. Working with variable scale data can be a problem, because a variable between 0 and 2000 will be more important in the training of the neural network than a variable between 0 and 7. This step will allow us to resize the distribution of the data on the same scale and eliminate the differences in magnitude between the features, thanks to the formula :

$$x'_{(j)} = \frac{x_{(j)} - \mu_{(j)}}{\sigma_{(j)}} \tag{4.1}$$

Where j is the index representing the feature among the five in our dataset, x' is the standardized value; x is the original value; μ is the mean value of the feature; σ is the standard deviation of the feature. This method can help improve the performance of our machine learning model by allowing a fairer comparison of features.

4.3.4 Split dataset

The original dataset is divided in two, a training dataset containing the 66% and a test dataset containing the remaining 33%. Generally, we find in the literature a division of 70% for the training dataset and 30% for the test dataset, we made the previous choice in order to be close to the literature, but also for the three anomalies to be in the test part.

4.3.5 Data preparation for an LSTM network

To study the degradation of a component of the system over time, the model must be able to have the capacity for long-term storage of significant data. Unlike conventional RNN (Recurrent Neural Network), the LSTM meets this need and also solves the vanishing and gradient exploding problems ([Hochreiter and Schmidhuber, 1997]), for more information see 2.2.4.2.

Stacked LSTM is a deep learning method consisting of several hidden layers, compared to Vanilla LSTM ([Wu et al., 2018]) which is limited to a single layer. This technique is very promising for hard-to-identify prediction cases ([Zhang et al., 2020]).

LSTM networks require as input multiple rows of the dataset which correspond to successive observations of the considered features. We fix the number of observation to 50 based on the work of [Chen et al., 2020]. The density meter observation that follows the 50 observations taken as input is considered as the desired output of the LSTM network.

In order to train the LSTM network, the input data needs to be organized in a specific way, which is typically a three-dimensional matrix (Figure 4.14). The first dimension of the matrix corresponds to the number of samples in the training dataset. The second dimension corresponds to the number of time steps in each sample. Finally, the third dimension corresponds to the number of features in each time step. For example, let's

say we have a training dataset consisting of 500 time steps, each with 5 features. We want to train an LSTM network to predict the density meter observations in the next time step, given previous 50 time steps of observations for each feature. In this case, the input data would need to be organized into a three-dimensional matrix with dimensions of $450 \times 50 \times 5$. Each horizontal slice of the matrix represents a single input of the LSTM network, which we call a sample. The first sample contains the first 50 observations of our five features, the second sample contains the 50 observations starting with the second observation, and so on until the last sample contains the 50 observations in our training dataset apart from the very last one. The training output data corresponds to the density meter observations starting from the 51st until the last one in the training dataset. In other words, the output data is a sequence of density meter observations that match the sequence of input samples. The LSTM network is trained to predict the next density meter observation in the sequence, given the previous 50 observations for each feature.

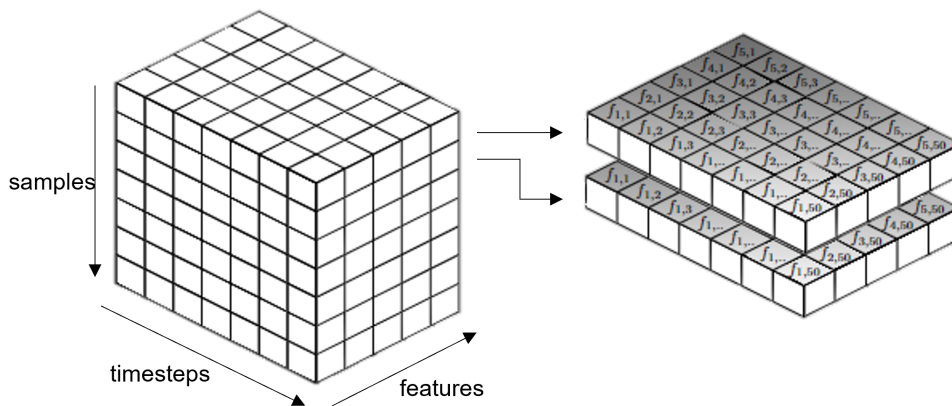


Figure 4.14 – Example Dataframe

4.3.6 Description of the LSTM model architecture

Our stacked LSTM model (Figure 4.15) consists of two layers, each containing an LSTM cell that is unrolled 50 times when the input of the 50 consecutive time steps of each sample is passed onto it, based on the work of [Chen et al., 2020] with only one Dense layer. The dimensionality of the input x_t at each time step t is equal to the number of features considered, i.e. 5 in our case, while the dimensionality of the hidden state of each LSTM cell is a parameter called units (see Table 4.4 for its tested values). The hidden state of the LSTM cell on the first layer is fed as input to the LSTM cell of the second layer. The hidden state of this latter cell, once all the data of the 50 time steps has been processed, is passed on to a Dense neuron which outputs the desired output, i.e. the predicted value of the density meter one time step after the considered 50 time steps. This value is finally compared to the actual value of the density meter in order to measure the prediction error and use it in order to tune the parameters of the stacked LSTM model.

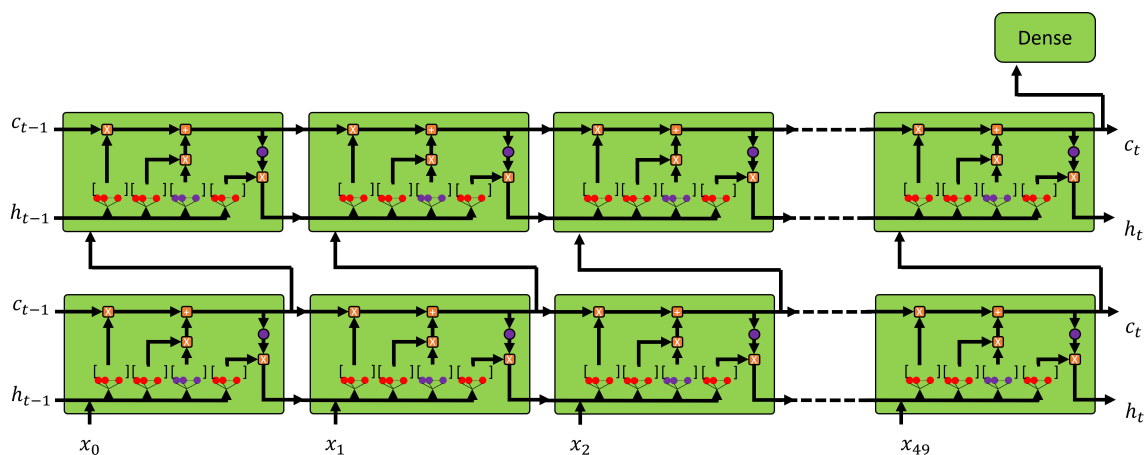


Figure 4.15 – LSTM model for one variable

Additional parameters that need to be tuned correspond to the dropout parameter, which we set to 0.2, and the batch-size and epochs which may be found in Table 4.4. The dropout technique in deep learning is widely used to prevent overfitting of neural network models. When using this technique, some neurons are randomly dropout, i.e. there are no more incoming and outgoing connections, during the forward propagation. Batch-size is used to split the training dataset into subsets and perform the training one batch at a time, while epochs represents the number of times the training procedure is repeated.

The best set of parameters is found using a grid-search approach, where all parameter combinations are tested and the one with the best performance is selected.

Table 4.4 – Grid search parameter values

Parameter Name	Value
Units	[90, 100, 110, 120]
Batch-size	[90, 100, 110, 120, 130]
epochs	[10, 20, 30, 40, 50]

4.4 Experimentation

4.4.1 Results

The results of our grid search give us the following best performing parameters on our dataset: units = 90, epochs = 10 and batch-size = 90.

We retrieve the model corresponding to this parameter setting and use it for anomaly detection by reconstructing the density variable over the test data. This corresponds to structuring the test dataset in the same way as the training dataset and then feeding the LSTM model on slice of the transformed dataset at a time. As each slice contains 50 consecutive observations of the 5 considered features, the model outputs the predicted subsequent observation of the density meter. This output is then compared to the actual value of the density meter for the same timestep in the test dataset. We illustrate the original density feature of the test dataset (in blue) and the reconstructed feature (in red) in Figure 4.16.

The LSTM has succeeded in reproducing the patterns of the test signal but with some deviations that will have consequences for identifying anomalies. We calculate the differences between the two signals and perform an outlier analysis method (see [Tukey et al., 1977]). This method considers that any value that is outside of $[Q1 - k \times IQR; Q3 + k \times IQR]$ is an outlier, with Q1 and Q3 being the first and third quartiles, IQR the interquartile range and $k=1.5$. Finally, we consider the 10% largest outliers as anomalies (see [Nam et al., 2020]). Figure 4.17 illustrates the anomalies over the original density meter measure.

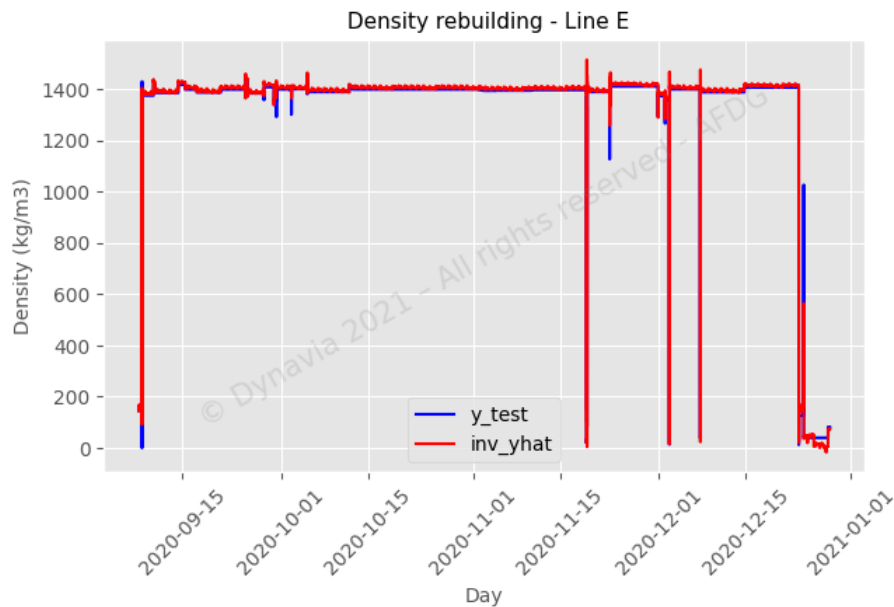


Figure 4.16 – Original and reconstructed density values over the test data - Line E

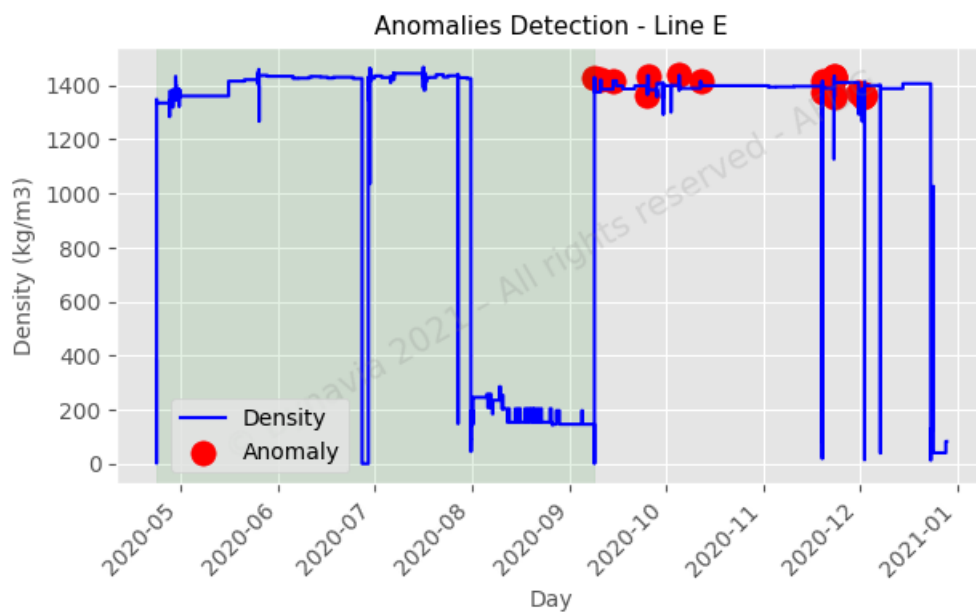


Figure 4.17 – Estimated dosing pump anomalies - Line E

The green area corresponds to the training data set, hence the absence of anomalies. Just after, we find the test data set, where we observe the anomalies proposed by our model. Table 4.5 gives more details on these anomalies.

- On 2020-09-08 at 1:13pm on the line loading problem (Figure 4.6), the model esti-

Table 4.5 – Estimated dosing pump anomalies - Line E

timestamp	value
2020-09-08 13:24:00	1428.715088
2020-09-10 09:35:30	1418,8427
2020-09-14 09:38:00	1416,0269
2020-09-25 06:17:00	1364,7043
2020-09-25 06:22:00	1435,0927
2020-10-05 05:39:00	1436,5260
2020-10-12 00:08:30	1414,8049
2020-11-19 08:50:30	1371.481812
2020-11-19 09:54:00	1415.759277
2020-11-23 01:41:00	1427,0262
2020-11-23 01:45:30	1424,8246
2020-11-23 03:07:00	1431,9301
2020-11-23 03:09:30	1366,8945
2020-11-23 04:06:00	1433,7494
2020-11-23 04:12:00	1357,3175
2020-11-30 21:20:00	1372,1207
2020-12-02 00:08:30	1365,5665
2020-12-02 00:11:00	1369,5217
2020-12-02 16:57:00	1361,3529

mated a single anomaly.

- On 2020-11-19 at 12:39pm, our model estimated two anomalies three/four hours before the maintenance case (Table 4.5), which can be interpreted as warning signs.
- Regarding the case of over-pressure and product leakage on 2020-12-03 at 4:45pm (Figure 4.7), the model estimates four anomalies the day before the failure.
- The other anomalies do not correspond to any maintenance ticket made by the customer. It is possible that these anomalies are real, however additional analysis of from other sensors and alarms would be needed.

4.4.2 Experimentations: LSTM with different configurations

The previous experience shows us that it is possible to identify anomalies, warning signs before a real failure of an industrial equipment. But is the current architecture and configuration of the model optimal? This section consists in answering this question through a multitude of experiments.

We observe, in Table 4.6, that by changing the "units" parameters of the LSTM network, the learning time remains almost the same within a few minutes. For a value of units equal to 10, we obtain the highest errors of the table and for a value of units equal to 90 we obtain the lowest errors. As for the R^2 , it remains the same. We can also see that with a number of units equal to 30, we obtain almost the same results as with a number of 90.

Table 4.6 – Model results and evaluations by changing the parameter units of the network a layer LSTM with a time steps of 50

units_input	start_time	duration	MAE	MSE	RMSE	R^2
10	18/04/2022 01:36:29	00:52:26	10,6522	457,3011	21,3846	0,9946
20	17/04/2022 11:11:03	01:00:39	7,1713	312,2172	17,6697	0,9963
30	17/04/2022 11:10:37	00:57:11	2,3033	225,0408	15,0014	0,9974
40	16/04/2022 13:57:36	00:57:09	6,2766	254,4660	15,9520	0,9970
50	16/04/2022 13:57:55	00:57:16	6,8169	280,5240	16,7488	0,9967
60	16/04/2022 12:51:46	01:01:40	8,4343	358,5907	18,9365	0,9958
70	16/04/2022 12:52:09	01:02:30	7,4657	267,7112	16,3619	0,9969
80	16/04/2022 11:29:33	01:02:01	9,0189	310,7860	17,6291	0,9964
90	16/04/2022 11:28:39	01:01:47	2,6311	216,4659	14,7128	0,9975

Let's now observe if a smaller time steps has an impact on the learning of the model, let's take as value 20.

Table 4.7 – Model results and evaluations by changing the parameter units of the network a layer LSTM with a time steps of 20

Nb_units_input	start_time	duration	MAE	MSE	RMSE	R2
10	20/04/2022 00:21:34	00:26:35	7,4867	333,9482	18,2742	0,9961
30	19/04/2022 23:51:23	00:28:04	2,9932	220,8473	14,8609	0,9974
50	19/04/2022 23:20:01	00:28:52	2,2112	212,5206	14,5781	0,9975
70	19/04/2022 22:46:53	00:29:37	5,4606	269,6162	16,4200	0,9968
90	19/04/2022 22:14:06	00:29:24	1,7440	209,7150	14,4815	0,9975

It is obvious that this choice of time steps of 20, in Table 4.7, has an impact on the learning time which is still divided by two. The R^2 value remains at 0.99 and also for a value of units equal to 10, we obtain the highest errors of the table and for a value of units equal to 90 we obtain the lowest errors. After all these experiments, which each time predict the same anomalies that correspond to the three real cases of maintenance, we find that the LSTM architecture with 1 single layer, a time step of 20 and a number of units at 50 allows to have the smallest errors for a training time of 28 minutes and 52 seconds

4.4.3 Experimentations: GRU with different configurations

Now that we have shown that an architecture with a single LSTM layer with a time step of 20 allows a learning time of 28 minutes and the smallest errors between the original signal and the predicted one, let's compare with a recurrent neural network the GRU (mentioned in section 2.2.4.4).

Table 4.8 – Model results and evaluations by changing the parameter units of the network a layer GRU with a time steps of 50

Nb_units_input	start_time	duration	MAE	MSE	RMSE	R2
10	22/04/2022 18:21:49	00:51:16	8,4075	286,0218	16,9122	0,9967
20	22/04/2022 15:25:48	00:52:22	8,9124	298,7234	17,2836	0,9965
30	22/04/2022 18:21:32	00:41:06	6,3333	256,8767	16,0274	0,9970
40	22/04/2022 14:37:20	00:42:46	9,8118	525,2137	22,9175	0,9939
50	22/04/2022 14:37:38	00:53:56	6,7784	253,5146	15,9221	0,9970
60	22/04/2022 13:26:21	00:46:28	6,2842	314,6602	17,7387	0,9963
70	22/04/2022 13:26:03	00:50:21	10,0942	314,6506	17,7384	0,9963
80	22/04/2022 11:08:58	00:49:15	11,2823	430,3840	20,7457	0,9950
90	22/04/2022 11:04:09	00:52:31	5,5057	247,4053	15,7291	0,9971

Comparing these results, Table 4.8, with the Table 4.6, we notice that between the two architectures the learning time is higher for the LSTM than for the GRU of maximum 12 minutes depending on the number of units. The errors have very close values in the cases of the MAE and RMSE on the other hand for the MAE there are big differences between the cases 10, 40, 80 units. In Table 4.9 the GRU layer with 10 units takes the least time to learn but has the largest errors. The configuration with 70 units has the highest learning time but the smallest errors.

Table 4.9 – Model results and evaluations by changing the parameter units of the network a layer GRU with a time steps of 20

Nb_units_input	start_time	duration	MAE	MSE	RMSE	R2
10	02/03/2023 12:54	00:19:37	9,6717	329,0345	18,1393	0,9962
30	01/03/2023 21:49	00:21:41	6,6244	253,6543	15,9265	0,9970
50	01/03/2023 21:49	00:25:30	5,9622	276,2826	16,6217	0,9968
70	01/03/2023 20:37	00:33:13	4,0026	222,0930	14,9028	0,9974
90	01/03/2023 20:30	00:29:32	4,3793	233,7373	15,2885	0,9973

Conclusion

The architecture of the model containing 2 layers of LSTM with 90 units, a time steps at 50, could be optimized by reducing it at first to one layer, which allowed to divide the learning time up to 4. Then by experimenting again on the time steps parameter, we observe that by reducing the time steps to 20 still with a number of units at 90 then we obtain a compromise with a learning time at 29 minutes and smallest errors while keeping an R^2 of 0.99. The GRU by its internal structure allows to have even smaller learning times, but the errors increase. Nevertheless, all architectures predict the same anomalies.

4.4.4 Experimentations: LSTM with pressure downstream

This experiment uses the LSTM architecture and configuration already performed for the density (see Table 4.10) by changing only the input variable: the downstream pressure where the overpressure and the product leakage occurred.

Table 4.10 – Model parameters with the pressure variable

Parameters	Details
Learning	Supervised
Nb records	3109810
Resample	30s
Split data	Training 66% and test : 33%
Number input units	90
Number hidden units	90
Number output unit	1
Activation function	Linear
Epochs	10
BatchSize	90
Time steps	50
Performance metrics	MAE, MSE, RMSE, R2

We observe (Figure 4.18) that the signal predicted by the model (in red) is uniform with an amplitude between 2-3 bar over the entire duration. The difference between the two curves is even greater when the pressure peaks are the largest in the period of 2020-12 and early 2021.

Evaluation metrics

Now, that our model has given predictive results with respect to the input data, we need to evaluate its predictions with the evaluation measurement tools in table 4.11 for a regression case seen in section 2.2.5.2. The LSTM model failed to reproduce the input signal patterns. The MAE, which is the average of the absolute differences between the predicted values (\hat{y}) and the actual values (y_{test}), of 0.218 indicates a rather small average error. But the RMSE of 0.654 indicates us that the average error is more important. The R^2 of 0.478 indicates that the model can be improved to better explain the variance of the dependent variable (y_{test}).

We also calculate the difference between the two signals and perform an IQR (In-

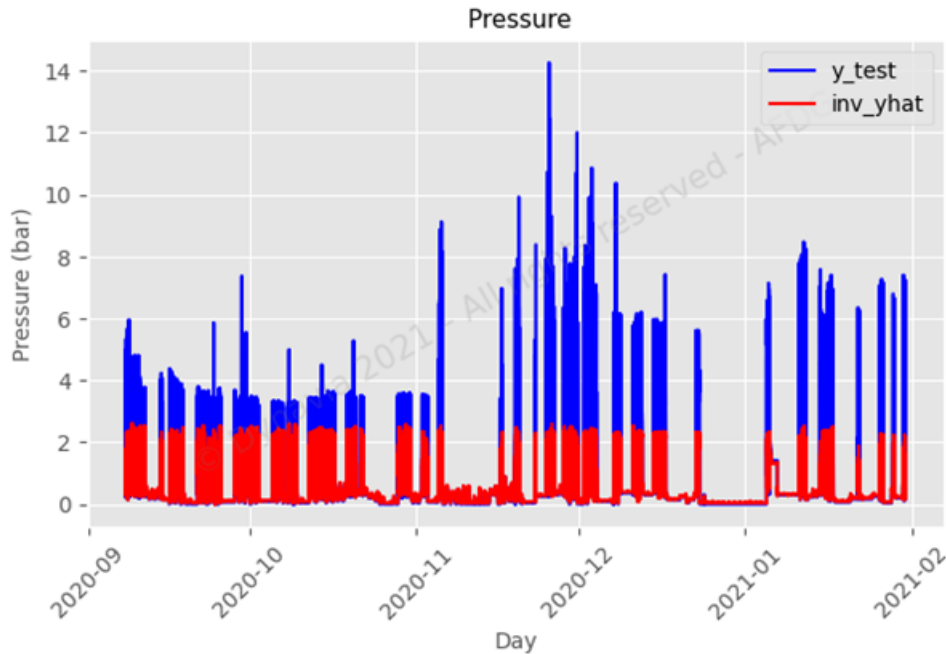


Figure 4.18 – Original and reconstructed pressure values over the test data

Table 4.11 – Metrics - LSTM model for pressure downstream

Metrics	Value
MAE	0.218
MSE	0.428
RMSE	0654
R ²	0.478

terquartile range) analysis method that identifies all data outside the interval $[Q1 - k \times IQR; Q3 + k \times IQR]$ as anomalies. Then we consider 10% of the most influential anomalies. Fig 4.19 shows the anomalies in the original pressure signal. The green area corresponds to the training dataset, hence the absence of anomalies. We obtain a dataframe with 77 anomalies over the period from 2020-11-05 to 2021-01-12.

By comparing this dataframe with the 3 failure cases (Table 4.2), we observe that there are no anomalies concerning the failure on the line loading on 2020-09-08 and the dosing problem on 2020-11-19. On the other hand, concerning the failure with the overpressure and the product leakage, we obtain 3 anomalies before the failure case on the same day (Table 4.12). The other anomalies do not correspond to known maintenance cases, but they may still be real, to be sure, we should continue with new analyses.

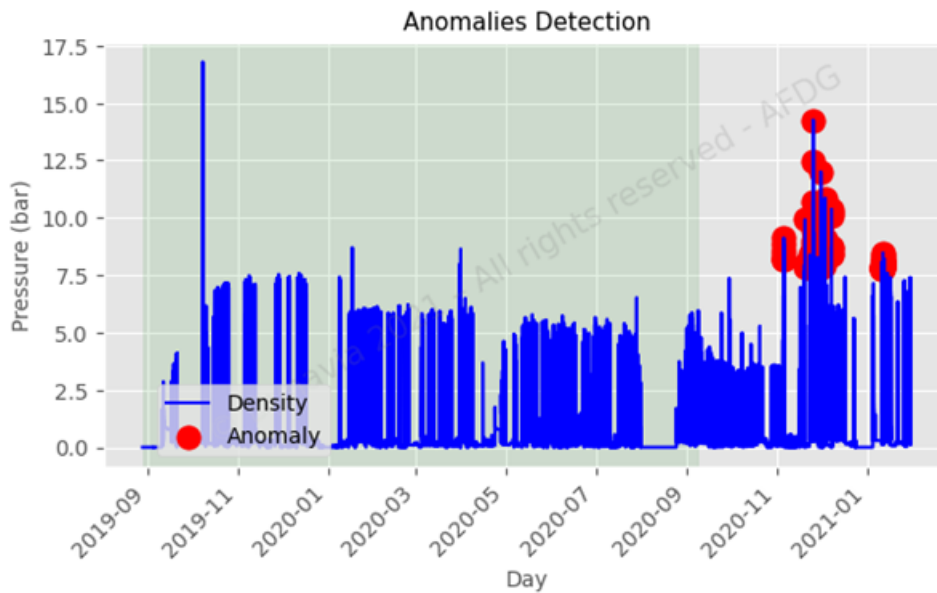


Figure 4.19 – Anomaly detection on the original signal - Line E

Table 4.12 – Metrics - LSTM model for pressure downstream

Timestamp	Value
2020-12-02 08:04:30	8.179
2020-12-02 11:10:00	7.889
2020-12-03 01:52:00	9.909
2020-12-03 12:43:00	9.099
2020-12-03 13:21:00	10.859

4.5 Comparison of results

The data provided in the Table 4.13 shows the results obtained from two different LSTM models, one with density and the other with vibration data, for identifying warning signs related to component failure. It is observed that the two models detected warning signs for the same third failure, which was related to overpressure problem with product leakage. This confirms the importance of using different models based on different data sources to identify anomalies that may lead to component failure.

Furthermore, it is interesting to note that the vibration data shows values that are very close in time to the actual failure, indicating that this data source could potentially be a useful input for predicting component failure.

Table 4.13 – Comparison results with density and vibration data using LSTM model.

Model	Timestamp	Value
LSTM with density	2020-12-02 00:08:30	1365,5665
	2020-12-02 00:11:00	1369,5217
	2020-12-02 16:57:00	1361,3529
LSTM with vibration	2020-12-02 08:04:30	8.179
	2020-12-02 11:10:00	7.889
	2020-12-03 01:52:00	9.909
	2020-12-03 12:43:00	9.099
	2020-12-03 13:21:00	10.859

4.6 Conclusion

In this work, we considered the problem of anomaly detection and used real data, density and pressure, from a metering pump used in a production facility. We focused on two failure cases by first performing manual analyses and comparing them with normal operating cases: line loading problem and overpressure with product leakage. We used models based on LSTM and GRU to detect these anomalies, with results showing that this can be done with fairly high accuracy, all the results of the experiments are available in the Appendix chapter in Figure 4.20. We were able to study the impact of the complexity of the LSTM and GRU architecture, i.e. by dividing the number of layers by two we divided the learning time by two. Then we also decreased the time step to 20 which also has an impact on the learning time which is again divided by two and finally we notice that this model with 90 units allows to have the smallest errors. Even if the LSTM has a more complex internet structure than the GRU, it allows in general to have smaller errors but both allow to have an optimal R^2 and they both detect the same anomalies. However, there are still several areas of improvement that we would like to explore in the future. These include applying the proposed method to other metering pumps and using other fluids as well as additional data from other sensors. Finally, the proposed approach can also be used to estimate the remaining lifetime of a device.

CONCLUSION AND PERSPECTIVES

Conclusion

With the rise of new technologies in Industry 4.0, the amount of data exploding, maintenance has become more intelligent and complex. The main objective of this thesis work was to implement a predictive maintenance strategy for the company Dynavia, in order to increase the availability of the system to increase its competitiveness.

Although preventive and corrective maintenance are always effective, predictive maintenance allows a good compromise between the two, but it remains difficult to implement with a high cost. Nevertheless, it is with this objective that in chapter 3, we propose a predictive maintenance strategy in an industrial context that allows, through increasingly precise analyses, to identify an ideal candidate.

This strategy allowed us to set up our experiment on the detection of anomalies in a dosing pump for a formulated product. Dynavia's maintenance department identified three anomalies: incorrect loading of a line with formulated product, incorrect dosing and finally overpressure with a product leak at the pump outlet.

A deep learning technique based on a recurrent neural network called Long Short Term Memory (LSTM) enabled us to identify the warning signs of the three real failures thanks to the product density data. A similar architecture but simplified by its internal structure, named Gated Recurrent Unit (GRU) also allowed us to find the same results, the difference is in the training time and in the performance metrics, but the LSTM with the architecture of a layer with a time step of 20 and a number of units 50 is considered the best in terms of learning time of 28 minutes and error evaluation.

Finally, we did an experiment, but this time with the downstream pressure data, the results show us that the model needs to be optimised with an r^2 of 48%, the model has difficulty in following the trend of the data, but it does allow us to find early warning

signs for pressure failure with product leakage. Thus, by running models in parallel on different data, one can certify that there are really anomalies and a real risk of failure.

Perspectives and future works

During the period of research and execution of my thesis, I was inspired and curious about several ideas in the field of predictive maintenance. Unfortunately, due to resource and time constraints, these ideas could not be fully developed. Nevertheless, this thesis offers new opportunities for future research work, not limited to:

- Due to the youth of the Dynavia equipment and the low number of actual failure cases, it was not possible to predict the remaining useful life of a specific component. In the future, it would be interesting to establish a failure history by using the surrounding data to build up a dataset. Then it would be possible to develop a new model and compare different regression algorithms to choose the one that offers the best performance, while ensuring maintainability.
- Set up the maintainability process of a model, test it and integrate it.
- It would be interesting to integrate the developed models into the studied Dynavia equipment in order to verify their operation and their ability to detect early signs of failure during production, with the aim of validating them.
- This work is focused on a single dosing pump with a single formulated product. An unanswered question is: can the developed models identify new anomalies with different pumps and different formulated products or should a specific model be created for each case?
- During the thesis, we also worked on the evaluation of the efficiency of a dosing pump with the aim of calculating its efficiency and studying its evolution over time, in order to detect any possible degradation. The objective was to set up a predictive maintenance process to predict this degradation and take decisions and maintenance actions if it exceeded a certain threshold, defined by the manufacturer. Unfortunately, we could not go beyond the literal stage, as we did not have access to the input power data of the pump motors. After discussions with the supplier, we were able to get this data, but this requires some computer development on the part of Dynavia, as well as an upgrade of the machine and a wait to get enough

data.

- It would be beneficial to extend this work to other Dynavia machines, as the question of the applicability of a model from one machine to another remains open. Indeed, this work was based on a single machine throughout the study, and it is important to verify whether the models developed are generalizable to other Dynavia machines.
- Although this thesis has addressed the research and technical aspects, the business aspect of implementing predictive maintenance at Dynavia has yet to be explored. A relevant question is whether Dynavia would first implement this approach in one of its machines to validate the concept, and whether it would consider offering predictive maintenance as a service to its customers, and if so, how this could be achieved.

Bibliography

- T Abbasi, King Hann Lim, NS Rosli, I Ismail, and R Ibrahim. Development of predictive maintenance interface using multiple linear regression. In *2018 International Conference on Intelligent and Advanced System (ICIAS)*, pages 1–5. IEEE, 2018.
- Mounia Achouch, Mariya Dimitrova, Khaled Ziane, Sasan Sattarpanah Karganroudi, Rizck Dhouib, Hussein Ibrahim, and Mehdi Adda. On predictive maintenance in industry 4.0: Overview, models, and challenges. *Applied Sciences*, 12(16), 2022. ISSN 2076-3417. doi: 10.3390/app12168081. URL <https://www.mdpi.com/2076-3417/12/16/8081>.
- Wasim Ahmad, Sheraz Ali Khan, MM Manjurul Islam, and Jong-Myon Kim. A reliable technique for remaining useful life estimation of rolling element bearings using dynamic regression models. *Reliability Engineering & System Safety*, 184:67–76, 2019.
- Mohiuddin Ahmed, Abdun Naser Mahmood, and Md. Rafiqul Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016. ISSN 0167-739X. doi: <https://doi.org/10.1016/j.future.2015.01.001>. URL <https://www.sciencedirect.com/science/article/pii/S0167739X15000023>.
- Ryan Ahmed, Mohammed El Sayed, S. Andrew Gadsden, Jimi Tjong, and Saeid Habibi. Automotive internal-combustion-engine fault detection and classification using artificial neural network techniques. *IEEE Transactions on Vehicular Technology*, 64(1):21–33, 2015. doi: 10.1109/TVT.2014.2317736.
- Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.
- Marcia Baptista, Shankar Sankararaman, Ivo. P. de Medeiros, Cairo Nascimento, Helmut Prendinger, and Elsa M.P. Henriques. Forecasting fault events for predictive maintenance using data-driven techniques and arma modeling. *Computers Industrial Engineering*, 115:41–53, 2018. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2017.10.033>. URL <https://www.sciencedirect.com/science/article/pii/S036083521730520X>.

-
- Joey Bernard. *Python Data Analysis with pandas*, pages 37–48. Apress, Berkeley, CA, 2016. ISBN 978-1-4842-0241-8. doi: 10.1007/978-1-4842-0241-8_5. URL https://doi.org/10.1007/978-1-4842-0241-8_5.
- Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- Marcus D. Bloice and Andreas Holzinger. *A Tutorial on Machine Learning and Data Science Tools with Python*, pages 435–480. Springer International Publishing, Cham, 2016. ISBN 978-3-319-50478-0. doi: 10.1007/978-3-319-50478-0_22. URL https://doi.org/10.1007/978-3-319-50478-0_22.
- Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- Alexandros Bousdekis and Gregoris Mentzas. Enterprise integration and interoperability for big data-driven processes in the frame of industry 4.0. *Frontiers in Big Data*, 4, 2021. ISSN 2624-909X. doi: 10.3389/fdata.2021.644651. URL <https://www.frontiersin.org/articles/10.3389/fdata.2021.644651>.
- George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- Andriy Burkov. *The hundred-page machine learning book*, volume 1. Andriy Burkov Quebec City, QC, Canada, 2019.
- Wahyu Caesarendra, Achmad Widodo, and Bo-Suk Yang. Application of relevance vector machine and logistic regression for machine degradation assessment. *Mechanical Systems and Signal Processing*, 24(4):1161–1171, 2010.
- Mikel Canizo, Enrique Onieva, Angel Conde, Santiago Charramendieta, and Salvador Trujillo. Real-time predictive maintenance for wind turbines using big data frameworks. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 70–77. IEEE, 2017.
- Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

-
- Chung Chen and Lon-Mu Liu. Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association*, 88(421):284–297, 1993.
- Zian Chen, Zhiyu Yan, Haojun Jiang, Zijun Que, Guozhen Gao, and Zhengguo Xu. Detecting coal pulverizing system anomaly using a gated recurrent unit and clustering. *Sensors*, 20(11), 2020. ISSN 1424-8220. doi: 10.3390/s20113271. URL <https://www.mdpi.com/1424-8220/20/11/3271>.
- Fuqiang Cheng, Xiaohong Guo, Yingying Qi, Jingwen Xu, Wan Qiu, Zhibao Zhang, Weitao Zhang, and Ningning Qi. Research on satellite power anomaly detection method based on lstm. In *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, pages 706–710, 2021. doi: 10.1109/ICPECA51329.2021.9362601.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. URL <https://arxiv.org/abs/1409.1259>.
- Francesca Cipollini, Luca Oneto, Andrea Coraddu, Alan John Murphy, and Davide Anguita. Condition-based maintenance of naval propulsion systems with supervised data analysis. *Ocean Engineering*, 149:268–278, 2018. ISSN 0029-8018. doi: <https://doi.org/10.1016/j.oceaneng.2017.12.002>. URL <https://www.sciencedirect.com/science/article/pii/S0029801817307242>.
- Harsh S. Dhiman, Dipankar Deb, S. M. Muyeen, and Innocent Kamwa. Wind turbine gearbox anomaly detection based on adaptive threshold and twin support vector machines. *IEEE Transactions on Energy Conversion*, 36(4):3462–3469, 2021. doi: 10.1109/TEC.2021.3075897.
- Sheng Ding, Andrey Morozov, Silvia Vock, Michael Weyrich, and Klaus Janschek. Model-based error detection for industrial automation systems using lstm networks. In Marc Zeller and Kai Höfig, editors, *Model-Based Safety and Assessment*, pages 212–226, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58920-2.
- Zhen Du, Lipeng Ma, Huakang Li, Qun Li, Guozi Sun, and Zichang Liu. Network traffic anomaly detection based on wavelet analysis. In *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 94–101, 2018. doi: 10.1109/SERA.2018.8477230.

J. Durkin. *Expert Systems Design and Development*. Macmillan Publishing Company, 1994.

Oche A Egaji, Tobore Ekwevugbe, and Mark Griffiths. A data mining based approach for electric motor anomaly detection applied on vibration data. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 330–334, 2020. doi: 10.1109/WorldS450073.2020.9210318.

Nabil El Malki, Franck Ravat, and Olivier Teste. Kd-means: clustering method for massive data based on kd-tree. In *22nd International Workshop On Design, Optimization, Languages and Analytical Processing of Big Data-DOLAP 2020-*, volume 2572. CEUR-WS, 2020.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

Len Feremans, Vincent Vercruyssen, Wannes Meert, Boris Cule, and Bart Goethals. A framework for pattern mining and anomaly detection in multi-dimensional time series and event logs. In Michelangelo Ceci, Corrado Loglisci, Giuseppe Manco, Elio Masciari, and Zbigniew Ras, editors, *New Frontiers in Mining Complex Patterns*, pages 3–20, Cham, 2020. Springer International Publishing. ISBN 978-3-030-48861-1.

Frank E Grubbs. Sample criteria for testing outlying observations. *The Annals of Mathematical Statistics*, pages 27–58, 1950.

Changchao Gu, Yihai He, Xiao Han, and Min Xie. Comprehensive cost oriented predictive maintenance based on mission reliability for a manufacturing system. In *2017 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–7. IEEE, 2017.

Tao Han, Yuqing Lan, Limin Xiao, Binyang Huang, and Kai Zhang. Event detection with vector similarity based on fourier transformation. In *2014 IEEE International Conference on Control Science and Systems Engineering*, pages 195–199, 2014. doi: 10.1109/CCSSE.2014.7224536.

Offshore Reliability Data Handbook. Oreda participants. *OREDA, PO Box, 370*, 2002.

-
- Fouzi Harrou, Ying Sun, and Sofiane Khadraoui. Amalgamation of anomaly-detection indices for enhanced process monitoring. *Journal of Loss Prevention in the Process Industries*, 40:365–377, 2016. ISSN 0950-4230. doi: <https://doi.org/10.1016/j.jlp.2016.01.024>. URL <https://www.sciencedirect.com/science/article/pii/S0950423016300225>.
- Zengyou He, Xiaofei Xu, Zhexue Joshua Huang, and Shengchun Deng. Fp-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems*, 2(1):103–118, 2005.
- Jean Heng. *Pratique de la maintenance préventive-4e éd.* Dunod, 2017.
- Marc Henrion, Daniel J Mortlock, David J Hand, and Axel Gandy. Classification and anomaly detection for astronomical survey data. In *Astrostatistical challenges for the new astronomy*, pages 149–184. Springer, 2013.
- Jordan Hochenbaum, Owen S. Vallis, and Arun Kejariwal. Automatic anomaly detection in the cloud via statistical learning, 2017. URL <https://arxiv.org/abs/1704.07706>.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
- M Hoseini, F Fegenabi, M Tajbakhsh, H Babazadeh-Igdir, et al. Introduction of seed treatment techniques (seed priming). *International Journal of Biosciences (IJB)*, 3(5): 1–12, 2013.
- Hexuan Hu, Bo Tang, Xuejiao Gong, Wei Wei, and Huihui Wang. Intelligent fault diagnosis of the high-speed train with big data based on deep neural networks. *IEEE Transactions on Industrial Informatics*, 13(4):2106–2116, 2017. doi: 10.1109/TII.2017.2683528.
- Mr RM Huddar and RV Kulkarni. Role of r and python in data science. *Res. JOURNEY*, page 32, 2018.
- International Organization for Standardization. Iso 22400-2:2014 – automation systems and integration – key performance indicators (kpis) for manufacturing operations management – part 2: Definitions, 2014. URL <https://www.iso.org/standard/60655.html>.

-
- Małgorzata Jasiulewicz-Kaczmarek, Stanislaw Legutko, and Piotr Kluk. Maintenance 4.0 technologies—new opportunities for sustainability driven maintenance. *Management and production engineering review*, 11, 2020.
- Ben Johnson and Anjana S Chandran. ‘comparison between python, java and r programming language in machine learning. *Int. Res. J. Modernization Eng. Technol. Sci.*, 3 (6):1–6, 2021.
- Nathalie Julien and Éric Martin. *L’usine du futur-Stratégies et déploiement-2e éd.: Industrie 4.0, de l’IoT aux jumeaux numériques*. Dunod, 2021.
- Ameeth Kanawaday and Aditya Sane. Machine learning for predictive maintenance of industrial machines using iot sensor data. In *2017 8th IEEE international conference on software engineering and service science (ICSESS)*, pages 87–90. IEEE, 2017.
- K Ramakrishna Kini and Muddu Madakyaru. Improved anomaly detection based on integrated multi-scale principal component analysis using wavelets: An application to high dimensional processes. *IFAC-PapersOnLine*, 53(1):398–403, 2020. ISSN 2405-8963. doi: <https://doi.org/10.1016/j.ifacol.2020.06.067>. URL <https://www.sciencedirect.com/science/article/pii/S2405896320300860>. 6th Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2020.
- Andrew Kusiak and Anoop Prakash Verma. Prediction of status patterns of wind turbines: A data-mining approach. *Journal of Solar Energy Engineering-Transactions of The Asme*, 133:011008, 2011.
- Guillaume Laloux. *Management de la maintenance selon l’ISO 9001: 2008*. Afnor, 2009.
- Chantal D Larose and Daniel T Larose. *Data science using Python and R*. John Wiley & Sons, 2019.
- Gilles Lasnier. Sureté de fonctionnement des équipements et calculs de fiabilités. pages 135–146. Hermès Science Publications, Paris, 2011. ISBN 9782746232013.
- Kwangsuk Lee, Jae-Kyeong Kim, Jaehyong Kim, Kyeon Hur, and Hagbae Kim. Cnn and gru combination scheme for bearing anomaly detection in rotating machinery health monitoring. In *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, pages 102–105, 2018. doi: 10.1109/ICKII.2018.8569155.

-
- Chuan Li, René-Vinicio Sanchez, Grover Zurita, Mariela Cerrada, Diego Cabrera, and Rafael E Vásquez. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical Systems and Signal Processing*, 76:283–293, 2016.
- Guannan Li, Huanxin Chen, Yunpeng Hu, Jiangyu Wang, Yabin Guo, Jiangyan Liu, Haorong Li, Ronggeng Huang, Hang Lv, and Jiong Li. An improved decision tree-based fault diagnosis method for practical variable refrigerant flow system using virtual sensor-based fault indicators. *Applied Thermal Engineering*, 129:1292–1303, 2018a. ISSN 1359-4311. doi: <https://doi.org/10.1016/j.applthermaleng.2017.10.013>. URL <https://www.sciencedirect.com/science/article/pii/S1359431117321580>.
- Hongkun Li, Yinhu Wang, Pengshi Zhao, Xiaowen Zhang, and Peilin Zhou. Cutting tool operational reliability prediction based on acoustic emission and logistic regression model. *Journal of Intelligent Manufacturing*, 26(5):923–931, October 2015. ISSN 0956-5515. doi: 10.1007/s10845-014-0941-4.
- Xiang Li, Qian Ding, and Jian-Qiao Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018b. ISSN 0951-8320. doi: <https://doi.org/10.1016/j.ress.2017.11.021>. URL <https://www.sciencedirect.com/science/article/pii/S0951832017307779>.
- Benjamin Lindemann, Nasser Jazdi, and Michael Weyrich. Anomaly detection and prediction in discrete manufacturing based on cooperative lstm networks. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1003–1010, 2020. doi: 10.1109/CASE48305.2020.9216855.
- M. Lochner and B.A. Bassett. Astronomy: Personalised active anomaly detection in astronomical data. *Astronomy and Computing*, 36:100481, jul 2021. doi: 10.1016/j.ascom.2021.100481. URL <https://doi.org/10.1016%2Fj.ascom.2021.100481>.
- Wei Lu and Ali A Ghorbani. Network anomaly detection based on wavelet analysis. *EURASIP Journal on Advances in Signal Processing*, 2009:1–16, 2008.
- Ruizhe Ma, Soukaina Filali Boubrahimi, Shah Muhammad Hamdi, and Rafal A Angryk. Solar flare prediction using multivariate time series decision trees. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2569–2578. IEEE, 2017.

-
- Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. Long short term memory networks for anomaly detection in time series. In *ESANN*, volume 2015, page 89, 2015.
- Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.
- R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.
- Brian Mwandau and Matunda Nyanchama. *Investigating Keystroke Dynamics as a Two-Factor Biometric Security*. PhD thesis, Strathmore University, 2018.
- Hong-Soon Nam, Youn-Kwae Jeong, and Jong Won Park. An anomaly detection scheme based on lstm autoencoder for energy management. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1445–1447, 2020. doi: 10.1109/ICTC49870.2020.9289226.
- Kie Ming Ngu, Nicholas Philip, and Shafishuhaza Sahlan. Proactive and predictive maintenance strategies and application for instrumentation amp; control in oil amp; gas industry. *International Journal of Integrated Engineering*, 11(4), Sep. 2019. URL <https://penerbit.uthm.edu.my/ojs/index.php/ijie/article/view/4193>.
- PJ García Nieto, E García-Gonzalo, F Sánchez Lasheras, and Francisco Javier de Cos Juez. Hybrid pso–svm-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability. *Reliability Engineering & System Safety*, 138:219–231, 2015.
- Ahmad Kamal Mohd Nor, Srinivasa Rao Pedapati, Masdi Muhammad, and Víctor Leiva. Overview of explainable artificial intelligence for prognostic and health management of industrial assets based on preferred reporting items for systematic reviews and meta-analyses. *Sensors*, 21(23), 2021. ISSN 1424-8220. doi: 10.3390/s21238020. URL <https://www.mdpi.com/1424-8220/21/23/8020>.
- Patryk Osypanka and Piotr Nawrocki. Resource usage cost optimization in cloud computing using machine learning. *IEEE Transactions on Cloud Computing*, 10(3):2079–2089, 2022. doi: 10.1109/TCC.2020.3015769.
- Ceyhun Ozgur, Taylor Colliau, Grace Rogers, Zachariah Hughes, and Elyse Bennie. Matlab vs. python vs. r. *Journal of data science: JDS*, 15:355–372, 07 2017.

-
- D. H. Pandya, S. H. Upadhyay, and S. P. Harsha. Fault diagnosis of rolling element bearing by using multinomial logistic regression and wavelet packet transform. *Soft Comput.*, 18(2):255–266, feb 2014. ISSN 1432-7643. doi: 10.1007/s00500-013-1055-1. URL <https://doi.org/10.1007/s00500-013-1055-1>.
- Daehyung Park, Yuuna Hoshi, and Charles C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018. doi: 10.1109/LRA.2018.2801475.
- Yong-chan Park, Jun-Gi Jang, and U Kang. Fast and accurate partial fourier transform for time series data. New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467293. URL <https://doi.org/10.1145/3447548.3467293>.
- J Phillips, Edward Cripps, John W Lau, and MR Hodkiewicz. Classifying machinery condition using oil samples and binary logistic regression. *Mechanical Systems and Signal Processing*, 60:316–325, 2015.
- Alberto Pliego Marugán, Ana María Peco Chacón, and Fausto Pedro García Márquez. Reliability analysis of detecting false alarms that employ neural networks: A real case study on wind turbines. *Reliability Engineering & System Safety*, 191:106574, 2019. ISSN 0951-8320. doi: <https://doi.org/10.1016/j.ress.2019.106574>. URL <https://www.sciencedirect.com/science/article/pii/S0951832018311918>.
- Akara Prayote. *Knowledge based anomaly detection*. PhD thesis, UNSW Sydney, 2007.
- Miguel Delgado Prieto, Giansalvo Cirrincione, Antonio Garcia Espinosa, Juan Antonio Ortega, and Humberto Henao. Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks. *IEEE Transactions on Industrial Electronics*, 60(8):3398–3407, 2012.
- PyPL Index. Popularity of programming language (pypl) index 2021, 2021. URL <https://pypl.github.io/PYPL.html>. Consulté le 9 février 2023.
- Zhaowei Qu, Lun Su, Xiaoru Wang, Shuqiang Zheng, Xiaomin Song, and Xiaohui Song. A unsupervised learning method of anomaly detection using gru. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 685–688, 2018. doi: 10.1109/BigComp.2018.00126.

-
- Mahathir Rahmany, Abdullah Mohd Zin, and Elankovan A Sundararajan. Comparing tools provided by python and r for exploratory data analysis. *IJISCS (International Journal of Information System and Computer Science)*, 4(3):131–142, 2020.
- Yongyi Ran, Xin Zhou, Pengfeng Lin, Yonggang Wen, and Ruilong Deng. A survey of predictive maintenance: Systems, purposes and approaches, 2019.
- Maria Riveiro, Mikael Lebram, and Marcus Elmer. Anomaly detection for road traffic: A visual analytics framework. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2260–2270, 2017. doi: 10.1109/TITS.2017.2675710.
- Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- Bernard Rosner. Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2):165–172, 1983.
- Douglas T Ross and Kenneth E Schoman. Structured analysis for requirements definition. *IEEE transactions on Software Engineering*, (1):6–15, 1977.
- Rafath Samrin and D Vasumathi. Review on anomaly based network intrusion detection system. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pages 141–147, 2017. doi: 10.1109/ICEECCOT.2017.8284655.
- Catherine D. Schuman, Thomas E. Potok, Robert M. Patton, J. Douglas Birdwell, Mark E. Dean, Garrett S. Rose, and James S. Plank. A survey of neuromorphic computing and neural networks in hardware, 2017. URL <https://arxiv.org/abs/1705.06963>.
- Uferah Shafi, Asad Safi, Ahmad Raza Shahid, Sheikh Ziauddin, and Muhammad Qaiser Saleem. Vehicle remote health monitoring and prognostic maintenance system. *Journal of advanced transportation*, 2018:1–10, 2018.
- Haidong Shao, Hongkai Jiang, Fuan Wang, and Huiwei Zhao. An enhancement deep feature fusion method for rotating machinery fault diagnosis. *Knowledge-Based Systems*, 119:200–220, 2017. ISSN 0950-7051. doi: <https://doi.org/10.1016/j.knosys.2016.12.012>. URL <https://www.sciencedirect.com/science/article/pii/S0950705116305068>.

-
- KK Sharma, US Singh, Pankaj Sharma, Ashish Kumar, and Lalan Sharma. Seed treatments for sustainable agriculture-a review. *Journal of Applied and Natural Science*, 7 (1):521–539, 2015.
- Rashmi B Shetty. Predictive maintenance in the iot era. *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*, pages 589–612, 2018.
- Deepak Sinwar and Manish Kumar. Anomaly detection using decision tree based classifiers. *International Journal of Modern Trends in Engineering and Research*, 3, 2016.
- Erotokritos Skordilis and Ramin Moghaddass. A condition monitoring approach for real-time monitoring of degrading systems using kalman filter and logistic regression. *International Journal of Production Research*, 55(19):5579–5596, 2017.
- Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, 2007. doi: 10.1109/TKDE.2007.1009.
- Greg Sullivan, Ray Pugh, Aldo P Melendez, and WD Hunt. Operations & maintenance best practices-a guide to achieving operational efficiency (release 3). Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2010.
- Wenjun Sun, Siyu Shao, Rui Zhao, Ruqiang Yan, Xingwu Zhang, and Xuefeng Chen. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement*, 89:171–178, 2016. ISSN 0263-2241. doi: <https://doi.org/10.1016/j.measurement.2016.04.007>. URL <https://www.sciencedirect.com/science/article/pii/S0263224116300641>.
- Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2015. doi: 10.1109/TII.2014.2349359.
- M’hamed Tahiri. *Informatisation de la maintenance GMAO/ERP, l’industrie 3.0 à l’ère de l’industrie 4.0*. Afnor Editions, 2021. ISBN 978-2-12-465790-2.

-
- David S. Thomas. The costs and benefits of advanced maintenance in manufacturing. NIST AMS 100-18, National Institute of Standards and Technology, p. NIST AMS 100-18, 2018. URL <https://doi.org/10.6028/NIST.AMS.100-18>.
- Gary L Tietjen and Roger H Moore. Some grubbs-type statistics for the detection of several outliers. *Technometrics*, 14(3):583–597, 1972.
- Tiobe Index. Top 20 programming languages 2021, 2021. URL <https://www.tiobe.com/tiobe-index/>. Consulté le 9 février 2023.
- Maximilian E. Tschuchnig and Michael Gadermayr. Anomaly detection in medical imaging – a mini review. 2021. doi: 10.48550/ARXIV.2108.11986. URL <https://arxiv.org/abs/2108.11986>.
- John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, Mass., 1977.
- Nishchal K. Verma, Vishal Kumar Gupta, Mayank Sharma, and Rahul Kumar Sevakula. Intelligent condition based monitoring of rotating machines using sparse auto-encoders. In *2013 IEEE Conference on Prognostics and Health Management (PHM)*, pages 1–7, 2013. doi: 10.1109/ICPHM.2013.6621447.
- Long Wang, Zijun Zhang, Huan Long, Jia Xu, and Ruihua Liu. Wind turbine gear-box failure identification with deep neural networks. *IEEE Transactions on Industrial Informatics*, 13(3):1360–1368, 2017. doi: 10.1109/TII.2016.2607179.
- Hadley Wickham and Garrett Grolemund. *R for data science: import, tidy, transform, visualize, and model data*. " O'Reilly Media, Inc.", 2016.
- Achmad Widodo, Eric Y. Kim, Jong-Duk Son, Bo-Suk Yang, Andy C.C. Tan, Dong-Sik Gu, Byeong-Keun Choi, and Joseph Mathew. Fault diagnosis of low speed bearing based on relevance vector machine and support vector machine. *Expert Systems with Applications*, 36(3, Part 2):7252–7261, 2009. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2008.09.033>. URL <https://www.sciencedirect.com/science/article/pii/S0957417408006611>.
- Dazhong Wu, Connor Jennings, Janis Terpenney, and Soundar Kumara. Cloud-based machine learning for predictive analytics: Tool wear prediction in milling. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2062–2069, 2016. doi: 10.1109/BigData.2016.7840831.

-
- Yuting Wu, Mei Yuan, Shaopeng Dong, Li Lin, and Yingqi Liu. Remaining useful life estimation of engineered systems using vanilla lstm neural networks. *Neurocomputing*, 275:167–179, 2018. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2017.05.063>. URL <https://www.sciencedirect.com/science/article/pii/S0925231217309505>.
- Takehisa Yairi, Yoshikiyo Kato, and Koichi Hori. Fault detection by mining association rules from housekeeping data. 06 2001.
- Jihong Yan and Jay Lee. Degradation Assessment and Fault Modes Classification Using Logistic Regression. *Journal of Manufacturing Science and Engineering*, 127(4):912–914, 07 2004. ISSN 1087-1357. doi: 10.1115/1.1962019. URL <https://doi.org/10.1115/1.1962019>.
- Deyong You, Xiangdong Gao, and Seiji Katayama. Wpd-pca-based laser welding process monitoring and defects diagnosis by using fnn and svm. *IEEE Transactions on Industrial Electronics*, 62(1):628–636, 2014.
- Jianbo Yu. Tool condition prognostics using logistic regression with penalization and manifold regularization. *Applied Soft Computing*, 64:454–467, 2018.
- Xiao yun Chen and Yan yan Zhan. Multi-scale anomaly detection algorithm based on infrequent pattern of time series. *Journal of Computational and Applied Mathematics*, 214(1):227–237, 2008. ISSN 0377-0427. doi: <https://doi.org/10.1016/j.cam.2007.02.027>. URL <https://www.sciencedirect.com/science/article/pii/S0377042707001100>.
- Lavina Yves. *Audit de la maintenance*. Editions d’Organisation, 1992. ISBN 978-2708114791.
- Bin Zhang, Shaohui Zhang, and Weihua Li. Bearing performance degradation assessment using long short-term memory recurrent network. *Computers in Industry*, 106:14–29, 2019a. ISSN 0166-3615. doi: <https://doi.org/10.1016/j.compind.2018.12.016>. URL <https://www.sciencedirect.com/science/article/pii/S0166361518304640>.
- Chen Zhang, Di Hu, and Tao Yang. Anomaly detection and diagnosis for wind turbines using long short-term memory-based stacked denoising autoencoders and xgboost. *Reliability Engineering & System Safety*, 222:108445, 2022. ISSN 0951-8320.

doi: <https://doi.org/10.1016/j.ress.2022.108445>. URL <https://www.sciencedirect.com/science/article/pii/S0951832022001107>.

Guicai Zhang, Changle Li, Haitao Zhou, and Timothy Wagner. Punching process monitoring using wavelet transform based feature extraction and semi-supervised clustering. *Procedia Manufacturing*, 26:1204–1212, 2018. ISSN 2351-9789. doi: <https://doi.org/10.1016/j.promfg.2018.07.156>. URL <https://www.sciencedirect.com/science/article/pii/S2351978918308321>. 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA.

Ke Zhang and Huidong Jin. An effective pattern based outlier detection approach for mixed attribute data. In Jiuyong Li, editor, *AI 2010: Advances in Artificial Intelligence*, pages 122–131, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-17432-2.

Qingqing Zhang, Jiyang Zhang, Jianxiao Zou, and Shicai Fan. A novel fault diagnosis method based on stacked lstm. *IFAC-PapersOnLine*, 53(2):790–795, 2020. ISSN 2405-8963. doi: <https://doi.org/10.1016/j.ifacol.2020.12.832>. URL <https://www.sciencedirect.com/science/article\protect\discretionary{\char\hyphenchar\font}{-}{}/pii/S2405896320311563>. 21st IFAC World Congress.

Weiting Zhang, Dong Yang, and Hongchao Wang. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 13(3):2213–2227, 2019b. doi: 10.1109/JSYST.2019.2905565.

Xiaoyuan Zhang, Yitao Liang, Jianzhong Zhou, and Yi zang. A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized svm. *Measurement*, 69:164–179, 2015. ISSN 0263-2241. doi: <https://doi.org/10.1016/j.measurement.2015.03.017>. URL <https://www.sciencedirect.com/science/article/pii/S0263224115001633>.

APPENDIX

Type	Number Layer	Freq Resample	Length Dataset	Time Steps	Nb units input	Nb units hidden1	Nb units output	Epochs	Batch Size	start time	end time	duration	MSE	RMSE	R2
LSTM	2	30s	417128	50	90	90	90	10	90	17/02/2023 14:06:14	17/02/2023 20:38:40	06:32:26	0.2180	0.4784	0.4794
LSTM	2	30s	321412	50	10	10	10	10	30	09/01/2023 18:13:45	10/01/2023 00:19:26	06:05:41	4.9473	324.2754	18.0076
LSTM	2	30s	321412	50	10	10	10	10	90	09/01/2023 13:47:33	09/01/2023 17:44:05	03:56:32	6.4944	401.0805	20.0270
LSTM	2	30s	321412	50	10	10	10	20	90	12/01/2023 08:41:56	12/01/2023 12:27:52	03:45:56	19.7066	638.6527	25.2716
LSTM	2	30s	321412	50	25	10	10	10	90	18/01/2023 09:30:28	18/01/2023 12:12:40	02:42:12	5.7251	303.3386	17.4166
LSTM	2	30s	321412	50	25	10	10	10	90	18/01/2023 09:30:28	18/01/2023 12:12:40	02:42:12	5.7251	303.3386	17.4166
LSTM	2	30s	321412	50	10	10	10	10	70	09/01/2023 17:00:16	09/01/2023 19:32:14	02:31:58	6.0864	312.9463	17.6903
GRU	1	30s	321412	40	90	90	90	10	80	01/03/2023 17:15	01/03/2023 18:20	01:04:20	3.6153	237.9420	15.4254
LSTM	2	30s	321412	50	10	10	10	10	80	03/01/2023 13:17:08	03/01/2023 15:41:31	02:24:23	15.9661	693.5516	26.3354
LSTM	2	30s	321412	60	10	10	10	10	90	31/12/2022 13:45:13	31/12/2022 15:55:36	02:10:22	4.0061	309.7171	17.5888
LSTM	2	30s	321412	60	10	10	10	10	90	31/12/2022 13:45:13	31/12/2022 15:55:36	02:10:22	4.0061	309.7171	17.5888
LSTM	2	30s	321412	70	10	10	10	10	90	31/12/2022 13:45:13	31/12/2022 15:55:36	02:10:22	4.0061	309.7171	17.5888
LSTM	2	30s	321412	80	10	10	10	10	90	31/12/2022 13:45:13	31/12/2022 15:55:36	02:10:22	4.0061	309.7171	17.5888
LSTM	2	30s	321412	50	90	90	90	10	90	27/12/2022 18:56:40	27/12/2022 19:24:59	00:28:19	4.1953	397.5737	19.9392
LSTM	2	30s	321412	50	90	90	90	10	90	15/04/2022 08:13:00	15/04/2022 10:28:08	02:15:07	11.4892	591.0049	24.3106
LSTM	2	30s	321412	50	70	70	70	10	90	15/04/2022 14:13:50	15/04/2022 16:24:09	02:10:19	8.1718	330.2992	18.1741
LSTM	2	30s	321412	50	80	80	80	10	90	15/04/2022 10:38:47	15/04/2022 12:47:14	02:08:27	7.1229	327.0396	18.0842
GRU	1	30s	321412	30	90	90	90	10	90	01/03/2023 18:59	01/03/2023 20:05	01:06:13	5.2917	254.2500	15.9452
LSTM	2	30s	321412	50	60	60	60	10	90	15/04/2022 16:35:24	15/04/2022 18:43:46	02:08:22	6.5507	380.5436	19.5075
LSTM	2	30s	321412	50	10	10	10	10	100	09/01/2023 13:17:16	09/01/2023 15:22:31	02:05:15	10.0127	393.3386	19.8328
LSTM	2	30s	321412	50	10	10	10	10	100	09/01/2023 13:17:16	09/01/2023 15:22:31	02:05:15	10.0127	393.3386	19.8328
LSTM	2	30s	321412	50	40	40	40	10	90	09/01/2023 13:17:16	09/01/2023 15:22:31	02:05:15	10.0127	393.3386	19.8328
LSTM	2	30s	321412	50	15	10	10	10	90	15/04/2022 22:36:34	16/04/2022 00:43:14	02:04:40	9.6764	438.4594	20.9384
LSTM	2	30s	321412	50	15	10	10	10	90	13/01/2023 16:52:15	13/01/2023 18:55:56	02:03:41	8.3550	335.6513	18.3208
LSTM	2	30s	321412	50	15	10	10	10	90	13/01/2023 16:52:15	13/01/2023 18:55:56	02:03:40	8.3550	335.6513	18.3208
LSTM	2	30s	321412	50	10	25	10	10	90	13/01/2023 09:06:13	13/01/2023 11:09:51	02:03:38	11.9500	520.4279	22.8129
LSTM	2	30s	321412	50	10	25	10	10	90	13/01/2023 09:06:13	13/01/2023 11:09:51	02:03:37	11.9500	520.4279	22.8129
LSTM	2	30s	321412	50	30	30	30	10	90	16/04/2022 04:51:18	16/04/2022 06:52:44	02:01:26	10.4312	514.4965	22.6825
LSTM	2	30s	321412	30	10	10	10	10	90	15/04/2022 22:39:53	16/04/2022 00:38:46	01:58:52	9.4413	395.9155	18.3280
LSTM	2	30s	321412	30	10	10	10	10	90	27/12/2022 19:36:10	27/12/2022 20:49:22	01:13:12	6.3203	441.5319	21.0127
LSTM	2	30s	321412	50	20	20	20	10	90	16/04/2022 04:51:46	16/04/2022 06:49:44	01:57:58	11.7067	398.2199	19.9554
LSTM	2	30s	321412	50	10	10	10	10	90	31/12/2022 13:34:07	31/12/2022 15:31:13	01:57:06	4.6126	377.9070	19.4398
LSTM	2	30s	321412	50	10	15	10	10	90	12/01/2023 15:31:27	12/01/2023 17:26:54	01:55:27	10.9820	592.5690	19.8134
LSTM	2	30s	321412	50	10	10	10	10	90	16/04/2022 09:24:01	16/04/2022 11:19:01	01:55:00	7.5575	378.6969	19.4601
LSTM	2	30s	321412	50	10	30	10	10	90	13/01/2023 14:47:56	13/01/2023 16:40:51	01:52:56	21.8601	858.7509	29.3045
LSTM	2	30s	321412	50	10	30	10	10	90	13/01/2023 14:48:18	13/01/2023 16:40:51	01:52:33	7.7972	338.7006	18.4038
LSTM	2	30s	321412	50	10	30	10	10	90	13/01/2023 14:48:18	13/01/2023 16:40:51	01:52:33	7.7972	338.7006	18.4038
LSTM	2	30s	321412	50	10	10	10	10	110	09/01/2023 17:00:04	09/01/2023 18:49:31	01:49:28	7.4941	387.0763	19.6743
LSTM	2	30s	321412	50	10	20	10	10	90	31/12/2022 16:01:40	31/12/2022 18:55:09	02:53:29	7.4387	508.1975	22.5432
LSTM	2	30s	321412	80	10	10	10	10	90	31/12/2022 16:01:40	31/12/2022 18:55:09	02:53:29	7.4387	508.1975	22.5432
LSTM	2	30s	321412	50	10	10	10	10	110	09/01/2023 17:00:04	09/01/2023 18:49:31	01:49:28	7.4941	387.0763	19.6743
LSTM	2	30s	321412	50	5	10	10	10	90	13/01/2023 16:52:05	13/01/2023 18:41:21	01:49:14	10.4433	496.7908	22.0633
LSTM	2	30s	321412	50	10	20	10	10	130	09/01/2023 09:02:35	09/01/2023 10:41:04	01:38:29	9.2919	357.0064	18.8847
LSTM	2	30s	321412	50	10	10	10	10	130	09/01/2023 13:47:28	09/01/2023 15:21:40	01:34:14	5.5973	383.6591	19.5872
LSTM	2	30s	321412	50	10	10	10	10	150	09/01/2023 13:47:28	09/01/2023 15:21:40	01:34:14	5.5973	383.6591	19.5872
LSTM	2	30s	321412	50	10	10	10	10	150	09/01/2023 15:28:05	09/01/2023 16:47:06	01:19:00	10.9888	499.4052	20.9620
LSTM	2	30s	321412	50	10	10	10	10	150	09/01/2023 15:28:05	09/01/2023 16:47:06	01:18:59	10.9888	499.4052	20.9620
LSTM	2	30s	321412	20	10	10	10	10	90	27/12/2022 17:46:23	27/12/2022 18:50:49	01:04:20	2.1368	286.9139	16.9385
LSTM	1	30s	321412	20	90	90	90	10	90	19/04/2022 22:14:06	19/04/2022 22:48:30	00:29:24	1.7440	209.7150	14.4815
LSTM	1	30s	321412	20	90	90	90	10	90	19/04/2022 22:14:06	19/04/2022 22:48:30	00:28:52	2.2112	212.5206	14.5781
LSTM	1	30s	321412	50	30	30	30	10	90	17/04/2022 11:20:37	17/04/2022 12:07:48	00:57:11	2.3033	225.0488	15.0014

LSTM	1	305	321412	50	90	90	1	10	90	16/04/2022 11:28:39	16/04/2022 12:30:26	01:01:47	2,6311	216,4659	14,7128	0,9975
LSTM	1	305	321412	20	30	30	1	10	90	19/04/2022 23:51:23	20/04/2022 00:19:27	00:28:04	2,9932	220,8473	14,8609	0,9974
GRU	1	305	321412	20	70	70	1	10	90	01/03/2023 20:37	01/03/2023 21:10	00:33:13	4,0026	222,0980	14,9028	0,9974
GRU	1	305	321412	20	90	90	1	10	90	01/03/2023 20:30	01/03/2023 21:00	00:29:32	4,3793	233,7373	15,2885	0,9973
LSTM	1	305	321412	20	70	70	1	10	90	19/04/2022 22:46:53	19/04/2022 23:16:30	00:29:37	5,4606	269,6162	16,4200	0,9968
GRU	1	305	321412	50	90	90	1	10	90	22/04/2022 11:04:09	22/04/2022 11:56:40	00:52:31	5,5057	247,4053	15,7291	0,9971
GRU	1	305	321412	20	50	50	1	10	90	01/03/2023 21:49	01/03/2023 22:15	00:25:30	5,9622	276,2826	16,6217	0,9968
LSTM	2	305	321412	90	10	10	1	10	90	31/12/2022 18:25:30	31/12/2022 21:40:13	03:14:43	9,8013	448,6913	21,1823	0,9948
LSTM	1	305	321412	50	40	40	1	10	90	16/04/2022 13:57:36	16/04/2022 14:54:45	00:57:09	6,2766	254,4660	15,9520	0,9970
GRU	1	305	321412	50	60	60	1	10	90	22/04/2022 13:26:21	22/04/2022 14:12:49	00:46:28	6,2842	314,6602	17,7387	0,9963
GRU	1	305	321412	50	30	30	1	10	90	22/04/2022 18:21:32	22/04/2022 19:02:37	00:41:06	6,3333	256,8767	16,0274	0,9970
GRU	1	305	321412	20	30	30	1	10	90	01/03/2023 21:49	01/03/2023 22:11	00:21:41	6,6244	253,6543	15,9265	0,9970
GRU	1	305	321412	50	50	50	1	10	90	22/04/2022 14:37:38	22/04/2022 15:31:34	00:53:56	6,7784	253,5146	15,9221	0,9970
LSTM	1	305	321412	50	50	50	1	10	90	16/04/2022 13:57:55	16/04/2022 14:55:12	00:57:16	6,8169	280,5240	16,7488	0,9967
LSTM	1	305	321412	50	20	20	1	10	90	17/04/2022 11:11:03	17/04/2022 12:11:41	01:00:39	7,1713	312,2172	17,6697	0,9963
LSTM	2	305	321412	100	10	10	1	10	90	31/12/2022 19:02:47	31/12/2022 22:35:36	03:32:49	10,8001	491,9742	22,1805	0,9942
LSTM	2	305	321412	100	10	10	1	10	90	31/12/2022 19:02:47	31/12/2022 22:35:36	03:32:49	10,8001	491,9742	22,1805	0,9942
LSTM	1	305	321412	50	70	70	1	10	90	16/04/2022 12:52:09	16/04/2022 13:54:39	01:02:30	7,4657	267,7112	16,3619	0,9969
LSTM	1	305	321412	20	10	10	1	10	90	20/04/2022 00:21:34	20/04/2022 00:48:09	00:26:35	7,4867	333,9482	18,2742	0,9961
GRU	1	305	321412	50	10	10	1	10	90	22/04/2022 18:21:49	22/04/2022 19:13:05	00:51:16	8,4075	286,0218	16,9122	0,9967
LSTM	1	305	321412	50	60	60	1	10	90	16/04/2022 12:51:46	16/04/2022 13:53:26	01:01:40	8,4343	358,5907	18,9865	0,9958
LSTM	2	305	321412	40	10	10	1	10	90	27/12/2022 20:54:11	27/12/2022 22:39:53	01:45:42	11,3624	472,7102	21,7419	0,9945
GRU	1	305	321412	50	20	20	1	10	90	22/04/2022 15:25:48	22/04/2022 16:18:11	00:52:22	8,9124	298,7234	17,2836	0,9965
LSTM	1	305	321412	50	80	80	1	10	90	16/04/2022 11:29:33	16/04/2022 12:31:35	01:02:01	9,0189	310,7860	17,6291	0,9964
GRU	1	305	321412	20	10	10	1	10	90	02/03/2023 12:54	02/03/2023 13:14	00:19:37	9,6717	329,0345	18,1393	0,9962
GRU	1	305	321412	50	40	40	1	10	90	22/04/2022 14:37:20	22/04/2022 15:20:05	00:42:46	9,8118	525,2137	22,9175	0,9939
GRU	1	305	321412	50	70	70	1	10	90	22/04/2022 13:26:03	22/04/2022 14:16:24	00:50:21	10,0942	314,6506	17,7384	0,9963
LSTM	1	305	321412	50	10	10	1	10	90	18/04/2022 01:36:29	18/04/2022 02:28:55	00:52:26	10,6522	457,3011	21,3846	0,9946
GRU	1	305	321412	50	80	80	1	10	90	22/04/2022 11:08:58	22/04/2022 11:58:13	00:49:15	11,2823	430,3840	20,7457	0,9950
LSTM	1	305	321412	40	90	90	1	10	90	01/03/2023 17:15:55	01/03/2023 18:20:15	01:04:20				

Figure 4.20 – All experiments on anomaly detection based on LSTMs and GRUs architectures.

Titre : Stratégie d'implémentation de la maintenance prédictive dans un contexte d'industrie 4.0 : une approche basée sur le machine learning

Mot clés : Stratégie Maintenance prédictive, détection d'anomalies, usine intelligente, apprentissage profond

Résumé : Cette thèse propose une stratégie de maintenance prédictive appliquée à un cas industriel avec Dynavia, une entreprise spécialisée dans la fabrication de systèmes de traitement des semences. Le système de Dynavia est innovant et assure la sécurité des opérateurs tout en permettant un traitement local sur les semences. Pour mettre en place la maintenance prédictive, un workflow a été établi pour identifier un composant idéal, collecter et préparer les données de plusieurs sources, créer un modèle de machine learning pour faire des prédictions et en assurer la maintenabilité lorsque le modèle se dégrade. Cette stratégie permet ainsi de mettre en place la maintenance prédictive pour ce projet industriel. Dans cette thèse, une expérimentation de la détection d'anomalie a été

réalisée sur une pompe de dosage industrielle à l'aide d'un algorithme de réseaux de neurones récurrents (LSTM et GRU) en utilisant la variable de la densité du produit dans la pompe. Différentes combinaisons d'architectures ont été testées pour identifier le modèle optimal. Les résultats montrent que tous les modèles testés ont détecté des anomalies similaires aux trois cas de maintenance réels déclarés par le client, avec des différences dans les écarts d'erreur d'évaluation des modèles et dans le temps d'apprentissage. Les modèles ont également été testés avec la pression aval de la pompe, et bien qu'il y ait place à l'optimisation, ils ont détecté des anomalies dans le troisième cas de réelle défaillance.

Title: Predictive maintenance implementation strategy in an industry 4.0 context: a machine learning based approach

Keywords: Predictive maintenance strategy, anomaly detection, smart factory, deep learning

Abstract: This thesis proposes a predictive maintenance strategy applied to an industrial case with Dynavia, a company specialized in the manufacture of seed treatment systems. Dynavia's system is innovative and ensures the safety of the operators while allowing a local treatment on the seeds. To implement predictive maintenance, a workflow was established to identify an ideal component, collect and prepare data from multiple sources,

create a machine learning model to make predictions and ensure maintainability when the model degrades. This strategy thus enables the implementation of predictive maintenance for this industrial project. In this thesis, an experiment of anomaly detection was performed on an industrial dosing pump using a recurrent neural network algorithm (LSTM and GRU) using the variable of the density of the product in the pump. Different combinations of ar-

chitectures were tested to identify the optimal model. The results show that all tested models detected anomalies similar to the three actual customer-reported maintenance cases, with differences in model evaluation error gaps and

learning time. The models were also tested with pump downstream pressure, and while there is room for optimization, they detected anomalies in the third real failure case.