POLITECNICO DI MILANO
Laurea Magistrale in Applied Statistics
Scuola di Ingegneria Industriale e dell'Informazione

DATA ANALYSIS FOR NUCLEAR POWER PLANTS MAINTENANCE : APPLICATION TO PNEUMATIC VALVES & CONTROL CLUSTERS

EDF
Saint-Denis, Francia

Relatore: Prof. Alessandra Menafoglio
Correlatore: Michel Guivarch

Tesi di Laurea Magistrale di:
Nahel Zidi, matricola 918965

Anno Accademico 2019-2020

Abstract

The use of data science is tending to become more widespread in industry and production. The present thesis was developed during an intership at EDF, the leading producer and supplier of electricity in France and Europe, within the UNIE-GMAP team, which is in charge of organising and optimising the maintenance of nuclear power plants, as well as defining maintenance strategies and programs. In this context, this work presents four independent data science missions. The first part concerns a survival analysis of pneumatic valve diaphragms. The objective of this mission is to carry out the survival analysis on all the membranes of the French nuclear park and then to possibly propose an optimal maintenance period for each type of pneumatic valve. The second mission concerns the control clusters, which are used to reduce the power or completely shut down a nuclear reactor. Here, the aim is to develop a functional data analysis tool that automatically examines a control cluster fall time curve, and detects whether the curve studied is within the norm or not and, if necessary, to carry out an initial diagnosis based on previously analysed curves. The third mission again deals with pneumatic valves, and represents a preliminary feasibility study of a tool to predict in advance an incident or malfunction of a valve by determining relevant indicators for the detection of such an event. Finally, the fourth and final mission concerns the rails used to guide the fall of the control clusters into the reactor, also known as cluster guides. The purpose of the study is, on the basis of the latest measurements made, to estimate the wear at the time of the next inspection so as to better quantify the number of cluster guides to be replaced. In this work, quantile regression forests are used, allowing us to improve the estimation accuracy with respect to the classical regression methods that have been used until now, and to cope with the very noisy nature of the data.

Key-words: survival analysis, functional data, outlier detection, classification, quantile regression forest

Sommario

L’uso della *data science* tende a diffondersi nell’industria e nella produzione. La presente tesi è stata sviluppata nel corso di un tirocinio presso EDF, il principale produttore e fornitore di energia elettrica in Francia e in Europa, all’interno del team UNIE-GMAP, che si occupa dell’organizzazione e dell’ottimizzazione della manutenzione delle centrali nucleari, nonché della definizione delle strategie e dei programmi di manutenzione. In questo contesto, questo lavoro presenta quattro missioni indipendenti di *data science*. La prima parte riguarda l’analisi della sopravvivenza dei diaframmi delle valvole pneumatiche. L’obiettivo di questa missione è di effettuare l’analisi di sopravvivenza su tutte le membrane del parco nucleare francese e di proporre eventualmente un periodo di manutenzione ottimale per ogni tipo di valvola pneumatica. La seconda missione riguarda i gruppi di controllo, che servono a ridurre la potenza o a spegnere completamente un reattore nucleare. In questo caso, l’obiettivo è quello di sviluppare uno strumento di analisi funzionale dei dati che esamini automaticamente una curva del tempo di caduta dei cluster di controllo, e rilevi se la curva studiata è nella norma o meno e, se necessario, di effettuare una diagnosi iniziale basata sulle curve analizzate in precedenza. La terza missione si occupa ancora una volta di valvole pneumatiche, e rappresenta uno studio preliminare di fattibilità di uno strumento per prevedere in anticipo un incidente o un malfunzionamento di una valvola, determinando indicatori rilevanti per la rilevazione di un tale evento. Infine, la quarta e ultima missione riguarda le rotaie utilizzate per guidare la caduta dei gruppi di controllo nel reattore, note anche come guide dei gruppi. Lo scopo dello studio è, sulla base delle ultime misurazioni effettuate, di stimare l’usura al momento della prossima ispezione in modo da quantificare meglio il numero di guide dei cluster da sostituire. In questo lavoro vengono utilizzate *quantile regression forests*, che permettono di migliorare l’accuratezza della stima rispetto ai metodi di regressione classici finora utilizzati, e di far fronte alla natura molto rumorosa dei dati.

Parole chiave: analisi di sopravvivenza, dati funzionali, rilevamento di anomalie, classificazione, quantile regression forest

Contents

List of Figures **9**

1 Introduction **14**

2 Valve membranes survival analysis **16**

2.1 Introduction 16

2.1.1 Background 16

2.1.2 Problem 18

2.1.3 Methodology 19

2.2 Theoretical background 19

2.3 Available data 21

2.4 Results 23

2.4.1 Kaplan-Meier estimators and feature groupings 23

2.4.2 Multivariate Weibull model 27

2.5 Discussion 30

3 Control cluster behaviours classification **31**

3.1 Introduction 31

3.1.1 Background 31

3.1.2 Problem 33

3.1.3 Functional data processing 33

3.1.4 Methodology 35

3.2 Theoretical background 36

3.2.1 Outlier curves detection algorithms 36

3.2.2 Functional K-nearest-neighbors algorithm 40

3.2.3 Functional K-means algorithm 43

3.3 Obtained results 43

3.3.1 Curves library creation 44

3.3.2 Outlier curves detection 46

3.3.3 Outlier classification 48

3.4 Discussion 53

4 ARE valves behaviour inventory 54

4.1 Introduction . . . 54

4.1.1 Background . . . 54

4.1.2 Methodology . . . 55

4.1.3 Data extraction . . . 55

4.2 Statistics monitoring over the country's nuclear power stations . . . 56

4.3 Evolution of the indicators before specific events . . . 61

4.4 Discussion . . . 65

5 Control cluster guides wearing estimate 66

5.1 Introduction . . . 66

5.1.1 Background . . . 66

5.1.2 Worn volume measurements . . . 68

5.1.3 Data cleaning . . . 69

5.2 Theoretical background . . . 72

5.2.1 Regression tree . . . 72

5.2.2 Random forests . . . 73

5.2.3 Random forest prediction intervals . . . 74

5.3 Prediction algorithms . . . 74

5.3.1 Linear regression . . . 74

5.3.2 Random forest . . . 78

5.4 Practical test . . . 82

5.5 Discussion . . . 83

6 Conclusion 84

Bibliography 86

A Appendix: Kaplan-Meier 88

List of Figures

2.1 Pneumatic actuator along with its membrane . . . 16
2.2 Variation of the membrane's position . . . 17
2.4 Torn membranes . . . 17
2.3 Deformed membrane . . . 18
2.5 Kaplan-Meier estimator (in years) for the first study perimeter . . . 18
2.6 Kaplan-Meier estimator for the whole study dataset . . . 23
2.7 Survival time boxplots . . . 24
2.8 Kaplan-Meier for each fluid value and the corresponding pairwise logrank test . . . 25
2.9 Kaplan-Meier estimator once the clustering done . . . 26
2.10 General fitted weibull survival curve . . . 27
2.11 Optimal predictive maintenance time obtained by summing up the hazard function and the preventive intervention rate . . . 28
2.12 Worst, average and best possible Weibull survival curves . . . 29
2.13 Optimal preventive maintenance time for the worst and the best cases . . . 30

3.1 Control rod . . . 31
3.2 The structure to which the rods are fixed: the spider . . . 31
3.3 Reactor in shutdown position: the control clusters are totally inserted . . . 32
3.4 Curve convertion algorithm input and output . . . 34
3.5 Smoothing parameters choice and obtained result . . . 35
3.6 Outliergram and functional boxplot applied to generated data . . . 40
3.7 Kernel functions . . . 42
3.8 Velocity curves before clustering . . . 44
3.9 K-means output for k=3 . . . 45
3.10 K-means output for k=4 . . . 46
3.11 Outliergram and functional boxplot applied to control cluster velocity curves . . . 47

3.12 Adjusted outliergram (5%) and functional boxplot applied to control cluster velocity curves 47
3.13 Number of neighbors choice cross-validation 49
3.14 Training set clusters. Blue curves correspond to usual shape curves while green and red ones correspond respectively to Issue1 and Issue2 50
3.15 Test set clusters. Blue curves correspond to usual shape curves while green and red ones correspond respectively to Issue1 and Issue2 52
3.16 Projection on the first two principal components computed on the training set. The colors are the same as the ones used to represent clusters. Crosses are the projection of the training set while dots are the test set's ones 52

4.1 ARE valve 54
4.2 One hour of raw data, moving average & mean difference 57
4.3 10 years extraction of both the indicators on a nuclear station 58
4.4 10 years extraction of both the standard deviation on a every P'4 nuclear station 59
4.5 10 years extraction of both the pseudo-frequency on a every P'4 nuclear station 59
4.6 Cullen-Frey graphs to determine the density to fit with 60
4.7 Fitted models for pseudo-frequency and standard deviation 61
4.8 Slow increasing of standard deviation and decreasing of pseudo-frequency during the two weeks preceding an incident 62
4.9 Two significant pseudo-frequency troughs before the incident . . . 63
4.10 Sudden change of both standard deviation and pseudo-frequency values a month before the incident 64
4.11 Case in which the indicators do no hint the final incident 64

5.1 Top of a control cluster 67
5.2 Mechanical structures guiding the control clusters during their insertion into the reactor 67
5.3 Measurements made when controlling the worn volume 68
5.4 2^{nd} measurement against the 1^{st} one 70
5.5 2^{nd} measurement against the 1^{st} one after data cleaning 71
5.6 ΔVol against ΔVol_{th} 76
5.7 Boxplots of ΔVol for each power plant 77
5.8 Variables importance 79
5.9 Predicted value along with their 90% prediction intervals 80

5.10 Centered true value along with their corresponding 90% prediction intervals 81

A.1 Kaplan-Meier estimator before and after clustering by circuit using pairwise Logrank test 88

A.2 Kaplan-Meier estimator before and after clustering by area of use using pairwise Logrank test 89

List of Algorithms

1 Outliergram algorithm 38

2 Outlier curves detection algorithm 39

3 Functional KNN algorithm 41

4 Functional weighted KNN algorithm 43

5 Functional K-means algorithm 44

6 Random forest learning algorithm 73

7 Random forest prediction algorithm 73

Acknowledgements

The internship opportunity I had with EDF was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it.

I express my deepest thanks to Michel Guivarch, my internship supervisor, for taking part in useful decision and giving necessary advices and guidance. Despite the containment situation, his accessibility and his in-depth knowledge of maintenance at EDF enabled me to carry out all the missions entrusted to me. I choose this moment to acknowledge his contribution gratefully.

I would also like to sincerely thank Sélim Benfeddoul, research engineer at EDF R&D, as well as Sophie Maingot, Philippe Paulin and Thierry Meylogan, tanks and control cluster specialists, Pierre Badel, researcher and fuel assembly project manager at EDF R&D, Julie Droniou and Victor Barrado, valves experts, Edwige Allain, statistician and cluster and assembly expert and finally Aurélie Arnaud, R&D researcher, for their precious help.

It is my radiant sentiment to place on record my best regards, my deepest gratitude to Prof. Alessandre Menafoglio for her careful and precious guidance which were extremely valuable for my work, both theoretically and practically.

Last but not least, I would like to deeply thank my parents and my brother with whom I spent most of my time in this particular context and who allowed me to concentrate on my work.

Chapter 1

Introduction

The use of data science is tending to become more widespread in industry and production. This is notably the case at EDF, the leading producer and supplier of electricity in France and Europe in which I did a 6-month internship. I was attached to the UNIE-GMAP team, which is in charge of organising and optimising the maintenance of nuclear power plants, as well as defining maintenance strategies and programs.

Until recently, maintenance-related strategy adjustments and optimisations were mainly based on feedback, but few large-scale data analysis studies were carried out for this purpose. To encourage the teams in charge of maintenance management to use these technologies in their work, project managers have been appointed in each of the concerned teams. Their aim is to make their colleagues aware of data analysis and to encourage them to train in this field.

It is in this context that I was able to carry out four independent data science missions. The structure of this dissertation is therefore in four parts completely independent of each other.

First part The first part concerns the survival analysis of pneumatic valve diaphragms. A previous study carried out on a reduced perimeter showed that the current preventive maintenance period could be increased, thus potentially reducing human and material needs. The objective of this mission is to carry out the survival analysis on all the membranes of the French nuclear park and then to possibly propose an optimal maintenance period for each type of pneumatic valve.

Second part The second mission concerns the control clusters, which are used to reduce the power or completely shut down a nuclear reactor. Until now, the analysis of the free-fall tests carried out on each of these clusters was done manually by an agent. This analysis consists of observing a speed versus time curve and detecting the presence or absence of anomalies. The objective of this mission is to develop a tool that automatically analyses a fall time curve, to detect whether the curve studied is within the norm or not and, if necessary, to carry out an initial diagnosis based on previously analysed curves.

Third part The third chapter again deals with pneumatic valves. Using a tool developed by R&D called Curiosity coupled with Python scripts, it is possible to extract the evolution over time of the flow rate measured by a sensor in the vicinity of any valve in the nuclear park. In doing so, it is possible to obtain a time series describing precisely the manoeuvres of each of the valves. Ultimately, the objective would be to develop a tool to predict in advance an incident or malfunction of a valve. In my case, the aim is to carry out a preliminary feasibility study of this tool by determining relevant indicators for the detection of such an event.

Fourth part Finally, the fourth and final mission concerns the rails used to guide the fall of the control clusters into the reactor, also known as cluster guides. The control clusters are coated with an anti-wear treatment that protects them from wear but leads to faster wear of the cluster guides. To ensure that the cluster guides continue to perform their function - i.e. that the clusters fit into the core quickly enough in the event that an automatic shutdown is required - this wear should not be too great. Therefore, EDF carries out periodic checks and, during these controls, replaces the cluster guides that are likely to be too worn by the next control.
The purpose of the study is, on the basis of the latest measurements made, to estimate the wear at the time of the next inspection so as to better quantify the number of cluster guides to be replaced. Feedback shows that estimating the volumes used on the basis of a single linear extrapolation of the last inspections leads to forecasting too many replacements and thus to mobilising unnecessary resources and unnecessarily lengthening unit outage schedules.
The main difficulty of this study lies in the data, which is very noisy.

For the sake of confidentiality, all the data presented in the following have been anonymized.

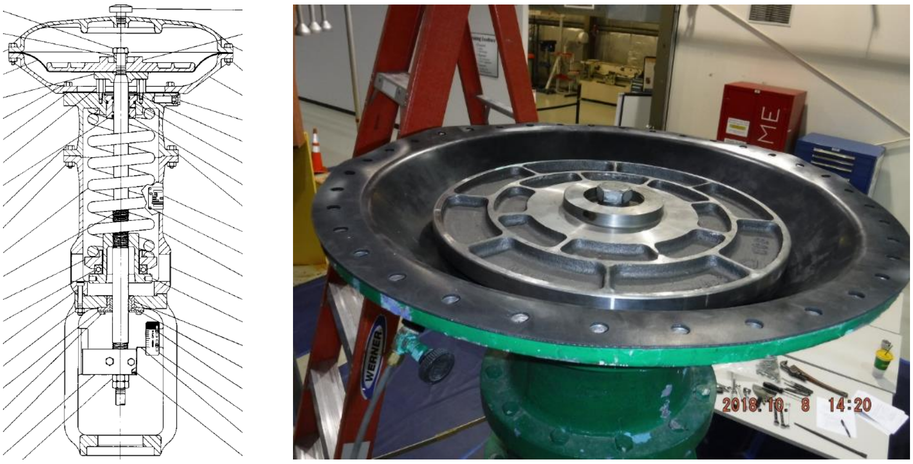
Chapter 2

Valve membranes survival analysis

2.1 Introduction

2.1.1 Background

On French nuclear power plants, the operation (opening/closing) of many valves is ensured by pneumatic actuators.



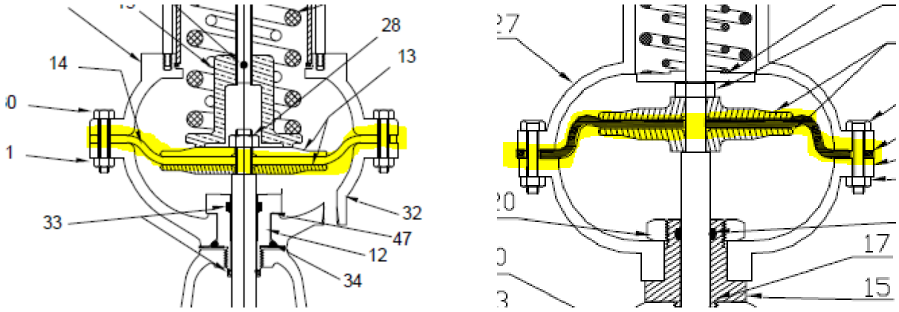
(a) Pneumatic actuator diagram

(b) Pneumatic actuator membrane

Figure 2.1: Pneumatic actuator along with its membrane

The opening and closing of the valve is fully controlled by the variation of air pressure. Some valves must be closed by default to guarantee the safety of the plant: the choice and the setting of the springs allows to have actuators that close the valve in case of loss of compressed air supply. In other cases, on the contrary, the valves must be open by default: a different choice of springs leads to their automatic opening in case of loss of compressed air supply to the actuator.

This pressure variation is allowed by the movement of an elastomer membrane, which is therefore solicited at each movement of the valve.

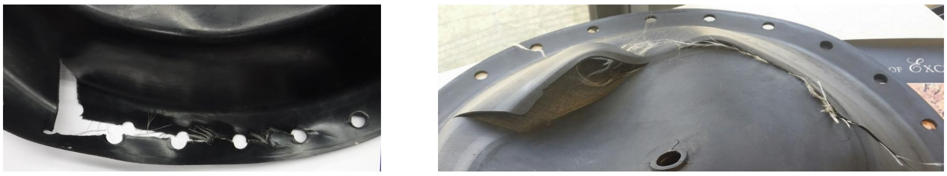


(a) Membrane in closed position

(b) Membrane in open position

Figure 2.2: Variation of the membrane's position

To date, a membrane is replaced approximately every ten years. In view of the large number of valves involved, the systematic replacement of these membranes represents a major economic challenge. The purpose of this study is to analyze their lifetime on the basis of maintenance actions carried out on these valves for about 25 years, with the aim, if possible, of optimizing the frequency of their replacement.



(a) Torn fasteners

(b) Torn central part of a membrane

Figure 2.4: Torn membranes

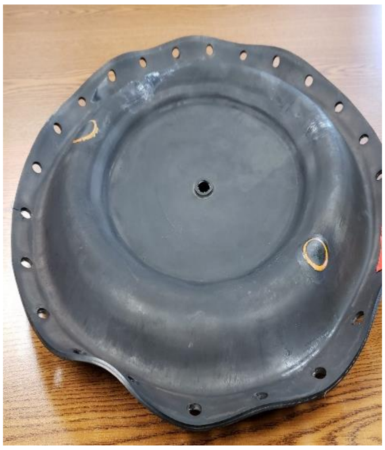


Figure 2.3: Deformed membrane

2.1.2 Problem

A first survival analysis had previously been carried out on a small perimeter containing approximately 200 valves over a period ranging from the commissioning of the corresponding nuclear power plants until today.

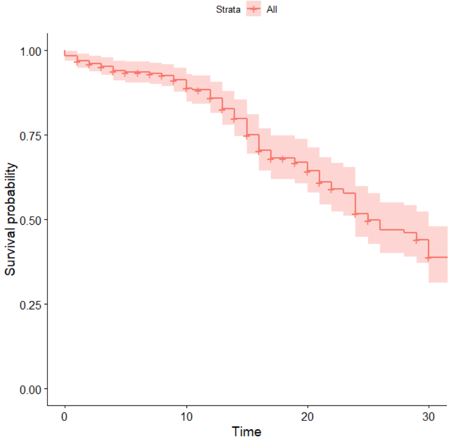


Figure 2.5: Kaplan-Meier estimator (in years) for the first study perimeter

This study shows that the membranes' fatigue is overstated, at least in the chosen perimeter, and the purpose of this first mission is to evaluate whether this result is global or not.

2.1.3 Methodology

The first part of the study will consist of redoing the study previously carried out on all the membranes of the nuclear park. Then, multivariate weibull model will be used to determine the optimal replacement frequency depending on various factors.

2.2 Theoretical background

To better understand the tools that will be used in this whole study, let us introduce some of the fundamental notions in survival analysis.

Definition. *For T the positive random variable representing time to event of interest, the **survival function** is defined as*

$$S(t) = P(T > t) = 1 - F(t)$$

*Then, let us define the **hazard function**, also known as the conditional failure rate, as*

$$h(t) = \frac{f(t)}{S(t)}$$

It describes the instantaneous risk that the event of interest happens within a very narrow time frame, and is commonly used to model which periods have the highest or lowest chances of an event.

Definition. *Censoring*

- ***Right censoring** occurs when a subject leaves the study before an event occurs or the study end before the event has occurred.*
- ***Interval censoring** occurs when the exact time of failure is not known; only an interval of time in which the failure occurred is recorded.*

Proposition. *A way of performing a survival analysis taking into account the different types of censoring introduced above is to define for each event a couple of times (T_1, T_2) such that:*

- $T_1 = T_2 < \infty$ *if the event is not censored*
- $T_1 < T_2 = \infty$ *if the event is right-censored with censoring time* $C = T_1$
- $T_1 < T_2 < \infty$ *if the event is interval censored*

One of the most commonly used tools to estimate the survival function is the so-called Kaplan-Meier estimator. It is defined as follows.

Definition. *Let us define the Kaplan-Meier estimate as*

$$\hat{S}(t) = \prod_{i:t_i \leq t} (1 - \hat{h}_i) = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

where $\hat{h}_i = \dfrac{d_i}{n_i}$ *is an estimate of the hazard function,* d_i *is the number of events that happened at time* t_i *and* n_i *the individuals known to have survived (have not yet had an event or been censored) up to time* t_i.

Definition. *Weibull distribution*
The probability density function of a Weibull random variable is:

$$f(x;\lambda,k) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \cdot e^{-\left(\frac{x}{\lambda}\right)^k} \mathbb{1}_{x\geq 0}$$

where $k > 0$ *is the shape parameter and* $\lambda > 0$ *is the scale parameter.*

The Weibull distribution is often used to model time-to-failure. In this case, the shape parameter k can be interpreted as follows :

- A value of $k < 1$ indicates that the failure rate decreases over time, i.e. there is a significant "infant mortality"
- A value of $k = 1$ indicates a constant failure rate over time
- A value of $k > 1$ indicates that the failure rate increases with time. It is often the case if there is an aging process.

The following definition is inspired by *The asymptotic properties of nonparametric tests for comparing survival distributions* by David Schoenfeld [11].

Definition. *The* ***logrank test*** *is a hypothesis test to compare survival distributions of two samples. It compares the hazard function of the two groupes at each observed event time.*
Let 1 and 2 be two groups of patients, $N_{1,t}$ *and* $N_{2,t}$ *be the number of subjects at risk at time t and* $O_{1,t}$ *and* $O_{2,t}$ *be the observed number of events in the groups at time t. Define* $N_t = N_{1,t} + N_{2,t}$ *and* $O_t = O_{1,t} + O_{2,t}$.
The null hypothesis is that the two groups have identical hazard functions, $H_0 : h_1(t) = h_2(t)$. *Under* H_0, *for each group* $i = 1,2$, $O_{i,t}$ *follows a hypergeometric distribution with parameters* $N_t, N_{i,t}, O_t$, *of mean* $E_{i,t}$ *and variance* $V_{i,t}$.

For all $t = t_1, ..., t_n$, the logrank statistic compares $O_{i,t}$ to its expected value under H_0 $E_{i,t}$. It is defined as

$$Z = \frac{\sum_{j=1}^{n}(O_{i,t_j} - E_{i,t_j})}{\sqrt{\sum_{j=1}^{n} V_{i,t_j}}} \xrightarrow{d} \mathcal{N}(0,1)$$

for $i = 1,2$

If the two groups have the same survival function, the logrank statistic is approximately standard normal.

2.3 Available data

The initial dataset is a file containing all maintenance operations having been made on the whole nuclear park's valves. Note that this file comes from an old database which is no longer active. Only some of them represent membrane replacements. For each of these operations, a data mining process has been implemented to extract from the written report whether the replacement was due to a routine visit or to the breakage of the membrane. This data mining step is a mandatory step considering the number of lines in this dataset but may have led to slightly noisy data.

As explained in the previous part, the censorship of the survival times of the membranes will be represented as intervals. After treatment and extraction of the right data, the study's dataset is as described in Figure 2.1.

RF	**Site**	**Unit**	**T1**	**T2**
RCP111VV	AAA	1	11	11
REN222VB	BBB	3	33	35
...	...	...	...	...

Table 2.1: Membrane failure dataset

The RF feature is equivalent to an ID and is related to another feature called RIN that encodes the valve type, the type of circuits it is linked to, the kind of fluid flowing through the circuit, etc. It will be used later in order to build different survival models depending on these parameters. In the same way, the Site and Unit features will be used to separate the different types of nuclear power plants that form the French nuclear park.

To better understand how T1 and T1 are computed, let us define for each nuclear power plant the following times:

- t_{start}: the commissioning year of the power plant
- t_{rec}: the commissioning year of the system recording all the events ($t_{start} \leq t_{rec}$)
- t_{end}: the year of the last recording in the database

The potential replacements that have occurred between the commissioning of the nuclear power plant and the commissioning of the database are unknown, this is why these two dates have to be taken into account.

The features T1 and T2 are built using the date of the maintenance events and different cases occur.
On the one hand, if the membrane replacement at time t corresponds to a breakage, then:

- if it is the first event for this valve, then $T1 = t - t_{rec}$ and $T2 = t - t_{start}$
- if it is at least the second event for this valve, then $T1 = t - t_{last\ event}$ and $T2 = T1$
- if it is the last event for this valve, then a row is added with $T1 = t_{end} - t$ and $T2 = \infty$

On the other hand, if the membrane replacement happens to be a routine replacement, then:

- if it is the first event for this valve, then $T1 = t - t_{rec}$ and $T2 = \infty$
- if it is at least the second event for this valve, then $T1 = t - t_{last\ event}$ and $T2 = \infty$
- if it is the last event for this valve, then a row is added with $T1 = t_{end} - t$ and $T2 = \infty$

Note that the rows for which $T2 < 2$ are considered as outliers since a diaphragm breaking happening less than two years after a maintenance is without a doubt due to a human error or to a manufacturing defect.

As introduced before, the whole dataset is now merged with the RIN dataset (that contains the valves' characteristics). Moreover, the Site feature is converted into a MWe class that describes the output power of each station. There exists 5 categories of it: CP0, CP1, CP2, N4 and P4. Once the features extracted from the RIN code, the dataset looks as Figure 2.2.

T1	T2	Circuit	Fluid	MWe class	Diameter	Designation	Area of use
11	11	AAA	VV	P4	80	C	T
33	35	BBB	VL	N4	100	X	X
...	...	...	...	...	...	...	...

Table 2.2: Final dataset

Note that the designation stands for the type of valve and the area of use for the pressure and temperature the valve may be exposed to.

2.4 Results

2.4.1 Kaplan-Meier estimators and feature groupings

In the following part, R package *icenReg* [1] is used to deal with interval censored data.

Let us first compute the Kaplan-Meier estimate for the whole dataset without taking into account the features.

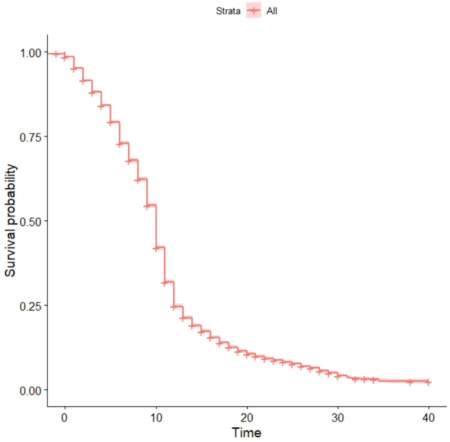


Figure 2.6: Kaplan-Meier estimator for the whole study dataset

The first impression here is that the result is significantly different with the one obtained in the original study. There might be several reason for which this difference is observed. Indeed, the first study perimeter is not a sample randomly chosen among the whole French nuclear park's valves. It is a sample containing mainly data that concerns valves located on secondary circuits. Therefore, two things are to emphasize. First of all, the chosen perimeter consists of valves that do not benefit from preventive maintenance. Then, since they belong to specific circuits, there may be a significant influence of both the circuit and the fluid that flows through it on the membrane's survival time.

To check this result, let us have a look at the distribution of the uncensored survival times according to the category they belong to. Figure 2.7 shows the boxplots for the 6 categorical features that are part of the study dataset.

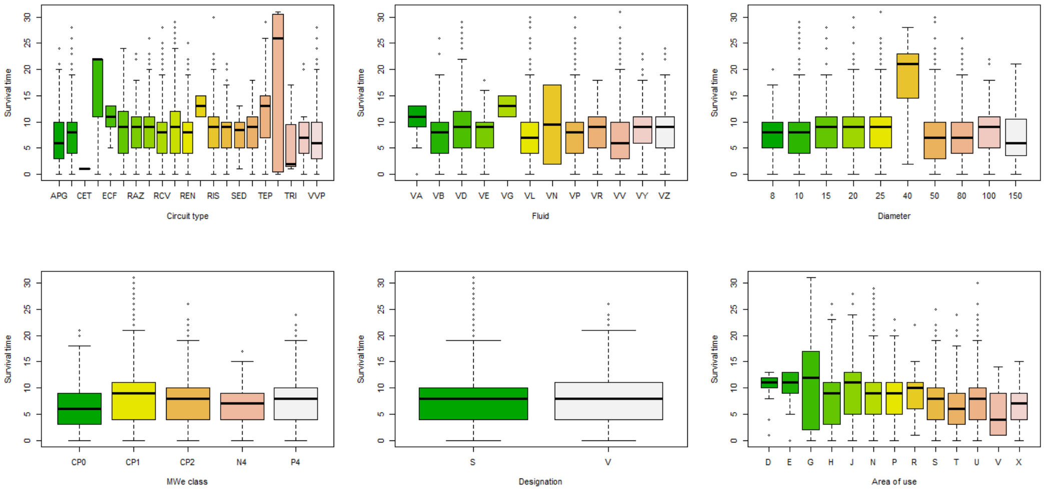
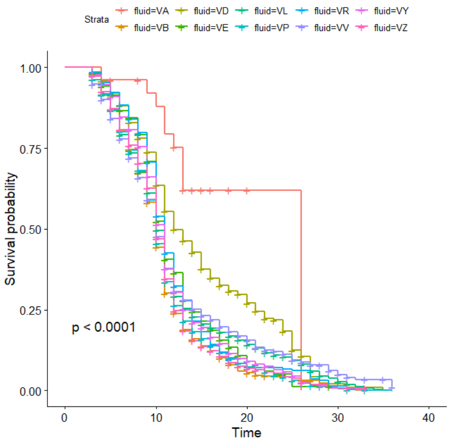


Figure 2.7: Survival time boxplots

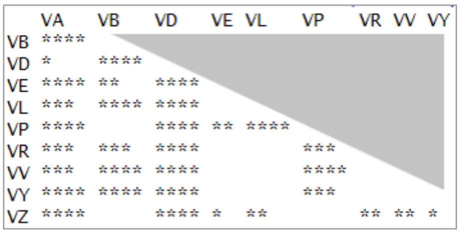
Several things can be observed on these boxplots. Firstly, median values of the survival times seem to be less than 10 in general. Then, some surprising results such as the low influence of the power level (MWe class), of the designation - which corresponds to the manufacturer - and of the diameter (except for D=40 which is a little represented value).

Only circuit, fluid and area of use features seem to have values that both stand out among other and have a sufficient amount of members to be taken into account.

Survival time against fluid feature Let us compute the Kaplan-Meier estimator for each value of the fluid feature. Note that the intervals are not plotted for more clarity.

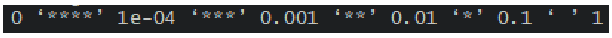


(a) Kaplan-meier estimator for each fluid value



	VA	VB	VD	VE	VL	VP	VR	VV	VY
VB	****								
VD	*	****							
VE	****	**	****						
VL	***	****	****						
VP	****		****	**	****				
VR	***	***	****			***			
VV	***	****	****			****			
VY	****	****	****			***			
VZ	****		****	*	**		**	**	*

(b) Pairwise logrank test output

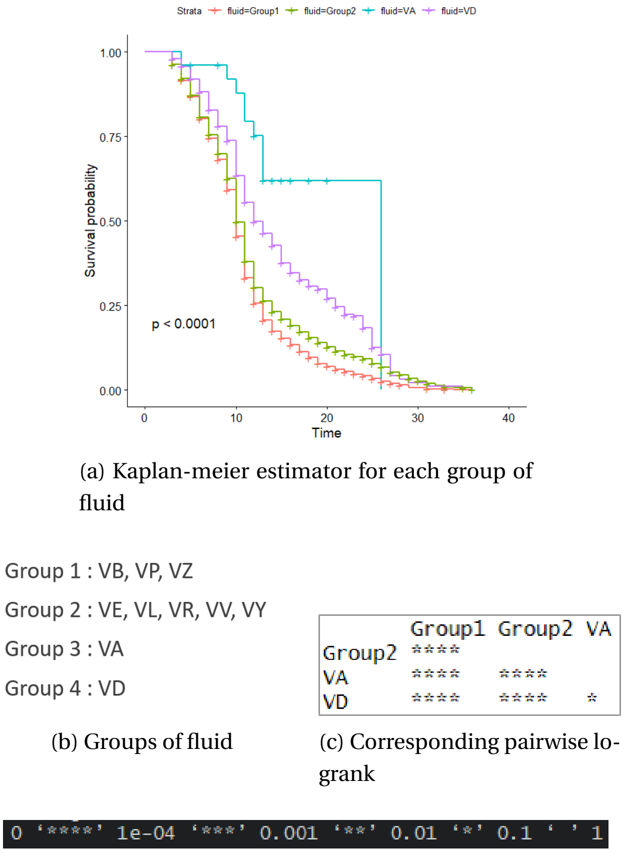


```
0 ‘****’ 1e-04 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.1 ‘ ’ 1
```

(c) Significance levels

Figure 2.8: Kaplan-Meier for each fluid value and the corresponding pairwise logrank test

Several curves stand out among the others and it is possible to distinguish between 2 and 4 groups of curves that share a similar shape. Then, using R package *pairwise_survdiff* [3], the pairwise logrank test gives an indication about whether two values of the fluid feature may or not be clustered together in a single group. Figure 2.8 shows the result of the corresponding test. The stars represent the p-value of the pairwise test. The more stars there are, the lower the p-value is and therefore the least these two values should be clustered together. Here, the choice is to gather two values of the fluid feature if the p-value is greater than 0.1, i.e if there is no star.



(a) Kaplan-meier estimator for each group of fluid

Group 1 : VB, VP, VZ

Group 2 : VE, VL, VR, VV, VY

Group 3 : VA

Group 4 : VD

(b) Groups of fluid

```
       Group1 Group2 VA
Group2 ****
VA     ****   ****
VD     ****   ****   *
```

(c) Corresponding pairwise logrank

```
0 '****' 1e-04 '***' 0.001 '**' 0.01 '*' 0.1 ' ' 1
```

(d) Significance levels

Figure 2.9: Kaplan-Meier estimator once the clustering done

Figures 2.8 and 2.9 show the Kaplan-Meier before and after the groupings. Once the first pairwise logrank test computed, each fluid value is grouped with the other values for which the test is conclusive, i.e the ones with which the test returns zero star. The obtained four groups are detailed on Figure 2.9.(b). Then, a new Kaplan-Meier estimate is computed as well as a new pairwise logrank test with the new fluid groups. This time, with the chosen threshold of 0.1, no grouping is needed and four groups remain.

Survival time against circuit and area of use features The same procedure has been applied to both circuit and area of use features. The results can be found in Appendix A, five circuit groups and four area of use ones are created in this way.

Now that the number of possible values for the triplet (Fluid, Circuit, Area of Use) has been reduced to $4 \cdot 5 \cdot 4 = 100$, a multivariate model can be built to estimate the optimal maintenance period for any valve's membrane.

2.4.2 Multivariate Weibull model

Global model Before taking into account the three previously mentioned covariates, let us first have a look at the maximum likelihood estimate of the membranes' survival function.

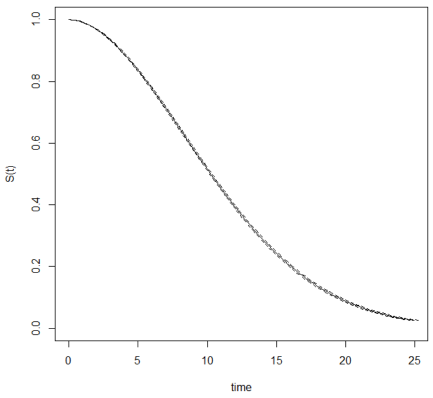


Figure 2.10: General fitted weibull survival curve

Scale parameter	**Shape parameter**
12.42	1.88

Table 2.3: Obtained Weibull parameters

The shape parameter is strictly greater than 1, which is an expected result. Indeed, in the case of a Weibull model, $k > 1$ indicates that the failure rate increases with time. This happens if there is an aging process i.e. if parts are more likely to wear out and/or fail as time goes. This is therefore entirely expected for elastomer membranes. Note that the scale parameter λ corresponds to the time at which 63% of the membranes are defective.

Then, the mean of the Weibull distribution corresponds to the mean time between failures (MTBF) and is computed as:

$$MTBF = \mu = \lambda \cdot \Gamma\left(1 + \frac{1}{k}\right)$$

with λ the scale parameter and k the shape parameter. Here, $MTBF \approx 11.03$ years. This supports the choice of a 10-year preventive maintenance period. An other way of using a Weibull model to predict the optimal predictive maintenance time is the one shown in Figure 2.11.

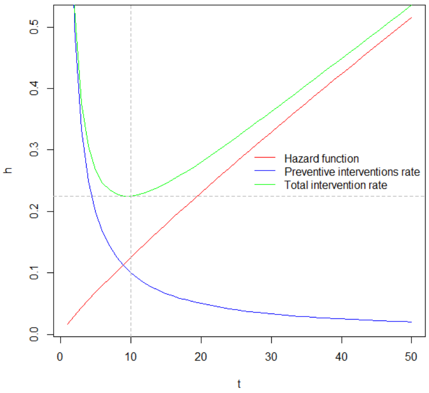


Figure 2.11: Optimal predictive maintenance time obtained by summing up the hazard function and the preventive intervention rate

The idea is to plot the hazard function corresponding to the fitted Weibull model and the proportion of preventive replacement. Since the scale is here in years, the hazard function represents the proportion of failure between t and $t+1$ while the proportion of preventive replacement is the function $t \mapsto \frac{1}{t}$. Summing up these two function, the proportion of replacements due to both failures and preventive maintenance is obtained. Then, minimizing this function gives an estimation of the optimal predictive maintenance time.

This time again, a value close to 10 years is obtained. Of course, this value has to be adjusted to deal with other constraints such as the fact that there exist light maintenance, during which an attempt is made to make a minimum of replacements, and its opposite, heavy maintenance.

Nonetheless, one could imagine having a preventive maintenance period that is specific for each valve or at least for each category of valves. This is the purpose of the following paragraph.

Model with covariates A Cox proportional hazard regression is used in this part. To use this, a strong assumption must be done. For each covariate, its effect must be independent of time. An easy way to verify it is to have a look at the Kaplan-Meier estimator for each variation of the considered covariates and the curves must not cross each other. With the groups that have been constituted for the three considered covariates, this test is passed.

Then, using the maximum likelihood estimator, a coefficient is returned for each value of each covariate. These coefficients β_{F_i}, β_{C_i} and β_{AoU_i} can be used to compute the conditional failure rate of a specific valve membrane. Not that β_{F_1}, β_{C_1} and β_{AoU_1} are fixed to zero since they correspond to the baseline. Then, knowing the groups a specific valve belongs to, its conditional failure rate is $h_{i,j,k}(t) = h(t) \cdot e^{\beta_{F_i}+\beta_{C_j}+\beta_{AoU_k}}$. The corresponding survival curves can be plotted as follows.

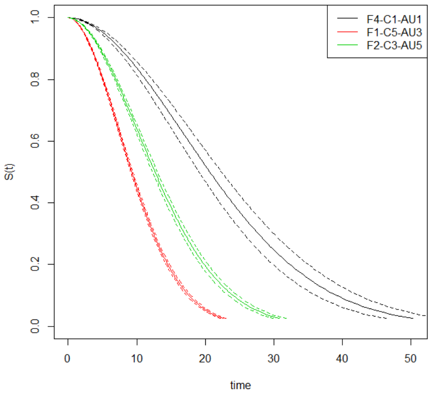
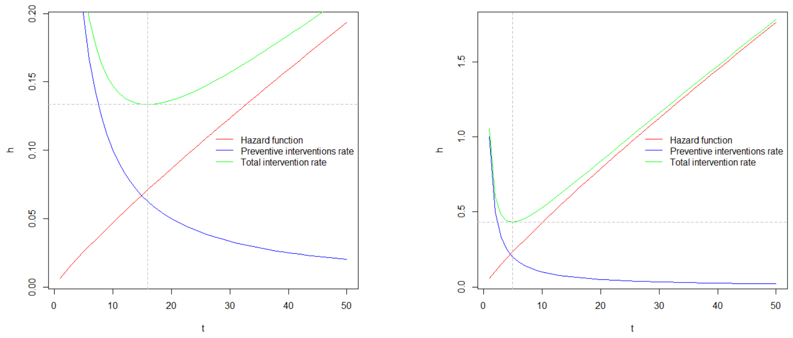


Figure 2.12: Worst, average and best possible Weibull survival curves

These three curves allow to understand that a global value of the maintenance period is not that easy to find. Although the value of 10 years that is currently used works well enough overall, an improvement would be to have a value of the maintenance period for each group of valves. Let us apply the previous graphical technique to the worst and the best combination of covariates values.



(a) 16 years optimal preventive time obtained in the best case

(b) 5 years optimal preventive time obtained in the worst case

Figure 2.13: Optimal preventive maintenance time for the worst and the best cases

Figure 2.13 confirms that there exists no absolute common optimal preventive time. Indeed, the gap between the membranes that are most in need of replacement and the ones least in need of replacement is of more that 10 years. Therefore, since maintenance cannot be carried out on a case-by-case basis since it requires the reactor to be shut down, one could imagine having groups of valves that are close in terms of maintenance time.

2.5 Discussion

Knowing that the dataset comes from an extraction work based on text mining algorithms, errors from this step could have been included in the data used. However, this amount of error is difficult to measure.

Nonetheless, survival analyses and Cox model allow a fine tuning of preventive maintenance. The application of the graphical methods confirms the choice of a 10-year predictive maintenance time.

However, it is possible to identify the equipment on which it would be possible to optimise the periodicity thanks to the Cox model. For example, in the case of a valve belonging to the F4-C1-AU1 groups, it would make sense to apply a maintenance time of approximately 16 years.

Chapter 3

Control cluster behaviours classification

3.1 Introduction

3.1.1 Background

On pressurized water reactors operated in France, the insertion of control clusters into the reactor is one of the two means used to control reactivity. These clusters are made up of 24 rods, themselves made up of a sheath containing neutrophage materials: inserting the bundles into the reactor thus makes it possible to capture neutrons that are no longer available for the nuclear reaction, which slows down the reaction. The 24 rods are fixed to a so called "spider" which is attached to a steel rod.

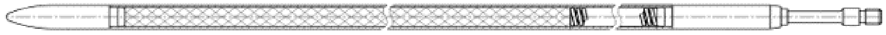


Figure 3.1: Control rod

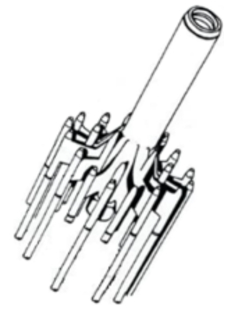


Figure 3.2: The structure to which the rods are fixed: the spider

The triggering of the fall of the control cluster is driven by an electromagnet. The total insertion of all the control clusters allows the shutdown of a reactor in less than 2 seconds. Nonetheless, various issues can prevent this whole operation from running properly. These potential issues are often deformations of the fuel assemblies or the swelling of the rods under irradiation.

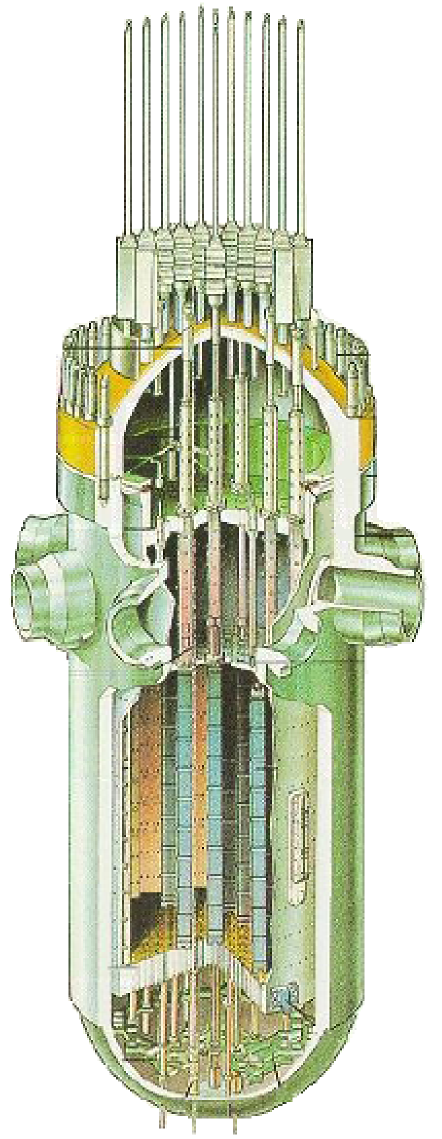


Figure 3.3: Reactor in shutdown position: the control clusters are totally inserted

Therefore, free falling tests are carried out on a regular basis to verify the proper functioning of this shutdown procedure. These tests consist in measuring the falling speed of the control clusters by letting them fall. The obtained data are curves of velocity against time that must be analyzed to detect possible anomalies.

3.1.2 Problem

Sometimes, some of these curves show some odd-looking patterns. The velocity of the corresponding control clusters is either too low, too high, oscillating a lot or having a very uncommon pattern while still respecting the two-seconds criterion. For now, when this kind of event occurs, an agent takes apart each abnormal curve and analyses it to determine whether the error comes from the acquisition or from a physical phenomenon on the control cluster and, in the second case, to identify the root of the problem. Therefore, the detection and analysis of atypical curves is subjective.

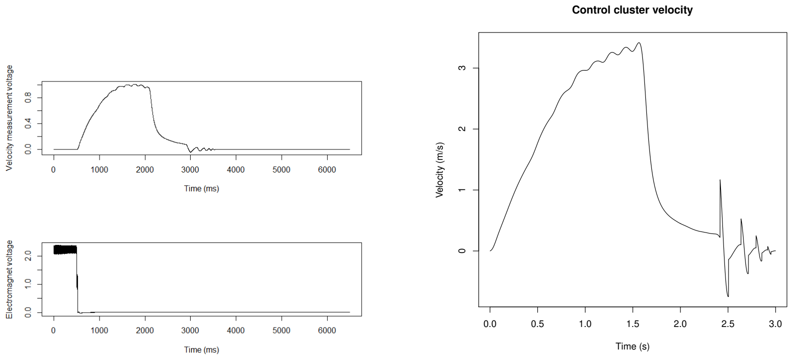
The aim of this mission is to develop a tool that is able to automatically detect an irregular curve and to give a first diagnosis on where the problem comes from. To do so, a combination of an outlier detection algorithm and a classification algorithm will be implemented.

3.1.3 Functional data processing

For every nuclear power plant, at each maintenance operation a folder is created containing the data recorded during the falling test.
Each raw curve is made of three columns:

- a time measurement (in s or ms depending on the file)
- a first voltage measurement proportional to the control cluster velocity
- a second voltage measurement which is non-zero when the electromagnet holding the control cluster is activated and zero when disabled

There is a proportionality factor between the first voltage measure and the control cluster velocity. Moreover, the time at which the electromagnet is being disabled can be interpreted as "zero time". With this in mind, a Python script is built; it first loads the raw curve, then set the zero time, convert the voltage into a velocity using the proportionality factor and finally adjusts the curve according to parameters that are the type of control cluster, the reactor power, etc. The obtained velocity curve contains 3000 points (1000 per second). It is lastly exported in a csv file.



(a) Raw data voltage curves

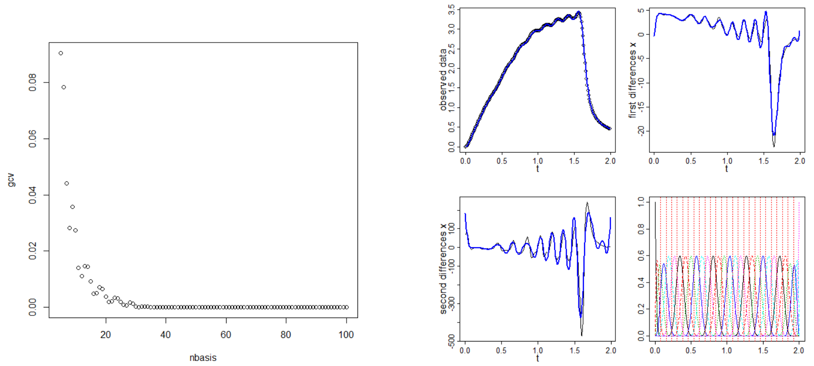
(b) Converted velocity curve

Figure 3.4: Curve convertion algorithm input and output

Some strong oscillations can be observed on the right hand side of the velocity curve of figure 3.4. These fluctuations correspond to the control cluster hitting the bottom of the tank and bouncing until equilibrium. Due to the high flexibility of the nuclear fuel rods and to the presence of springs between the control cluster and the control stem, this part of the signal is extremely complex to deal with. Therefore, the study will mainly focus on the evolution of velocity before this fluctuating part.

To get rid of this oscillating part, an indicator called T_5 is used. It corresponds to the time at which the velocity curve reaches its maximum. When it comes to analyzing a curve dataset, all the curves are truncated at $t = \tilde{T}_5 + 0.6$ seconds, where $\tilde{T}_5$ denotes its median. This choice is arbitrary but shows good performances since it removes any late oscillation without avoiding the user to analyze slow control cluster curves.

Once the truncation done, a last issue has to be dealt with. The raw data voltage curves acquisition is often disrupted by a 50 Hz noise. This noise is largely removed by the Python script that converts the raw curves into velocity curves. However, the obtained curves are too heavy to be analyzed quickly. In order to deal with this issue, the idea is to sample and then smooth each curve. Each curve is sampled with a sampling frequency of 100 Hz. In this way, the curves csv files are lightened and no important information is lost. Then, using splines of order 5, the general cross-validation plotted in Figure 3.5 is computed.



(a) GCV as a function of the number of basis

(b) Smoothed velocity curve for nbasis=30 along with its first and second derivatives

Figure 3.5: Smoothing parameters choice and obtained result

An elbow is easily observable for a number of splines nbasis ≈ 30. On the right of the figure, a smoothing attempt with nbasis= 30 is plotted. The smoothed curve perfectly fits the original one but no overfitting can be observed. The first and second derivative curves are indeed similar for both the smoothed and the original data.

3.1.4 Methodology

This problem resolution consists of two distinct parts.

First of all, the building of a sample of known curves that contains both curves with expected shapes and curves with abnormal shapes whose problem is well-known. This step consists of classifying them into clusters using a K-means algorithm. This classification is then refined by a domain expert. At this stage, he is in charge of labelling each cluster and/or batching them into larger clusters if they correspond to similar types of curves. Once this building step is done, the so-called "curves library" is filled with every clustered curves as well as a file that contains their classification. Their corresponding class is either the class of expected shape curves or the physical problem that is directly linked to the shape they have.

Then, using an outlier detection algorithm along with a classification algorithm, any new control cluster velocity curve dataset can be classified. Here are the several steps of the whole algorithm:

1. A new dataset is created, composed of both the known **normal shape** curves and the new curves to classify

2. For each new curve, if the outlier detection algorithm detects the new curve as an outlier, go to step 3. Otherwise, classify the curve as a normal curve.

3. Perform a supervised machine learning algorithm, using as training set the **whole** sample of known curves and their associated classes, on the detected outlier curves

3.2 Theoretical background

3.2.1 Outlier curves detection algorithms

This part refers to *Shape Outlier Detection and Visualization for Functional Data: the Outliergram* by Ana Arribas-Gil & Juan Romo [2].

Let us first introduce two metrics that will be used to determine whether a curve's shape is usual or not among a functional dataset.

Definition. *Modified Band Depth*

$$MBD_{\{x_1,\ldots,x_n\}}(x) = \binom{n}{2}^{-1} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{\lambda(\{t \in I | \min(x_i(t), x_j(t)) \leq x(t) \leq \max(x_i(t), x_j(t))\})}{\lambda(I)}$$

For a given curve $\mathscr{C}$, the MBD index measures the mean proportion of "time" spent by $\mathscr{C}$ in between two curves. For a curve that lays upward or below all the others, $MBD(\mathscr{C}) = 0$. If a curve is the curve precisely located at the center of all the functional dataset, then, $MBD(\mathscr{C}) = 0.5$ i.e. the maximum is reached.

Definition. *Modified Epigraph Index*

$$MEI_{\{x_1,\ldots,x_n\}}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{\lambda(\{t \in I | x_i(t) \geq x(t)\})}{\lambda(I)}$$

For a given curve $\mathscr{C}$, the MEI index measures the mean proportion of "time" spent by $\mathscr{C}$ below each other curves. Therefore, $MEI(\mathscr{C}) = 0$ for a curve upward every other ones and $MEI(\mathscr{C}) = 1$ for a curve below every other ones.

Proposition. *It can be proved that*

$$MBD_{\{x_1,\ldots,x_n\}}(x) = a_0 + a_1 MEI_{\{x_1,\ldots,x_n\}}(x) + a_2^2 MEI_{\{x_1,\ldots,x_n\}}(x)^2 \\ + a_2 \left[\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{\lambda(E_{i,x} \cap E_{j,x})}{\lambda(I)} - \frac{\lambda(E_{i,x})\lambda(E_{j,x})}{\lambda(I)^2} \right) \right]$$

with $a_0 = a_2 = \dfrac{-2}{n(n-1)}$, $a_1 = \dfrac{2(n+1)}{n-1}$ *and* $E_{i,x} = \{t \in I | x_i(t) \geq x(t)\}$.

Corollary. *For any sample* $x_1, \ldots, x_n$ *of continuous functions on* I*, it holds that for any* $x \in \{x_1, \ldots, x_n\}$*,*

$$MBD_{\{x_1,\ldots,x_n\}}(x) \leq a_0 + a_1 MEI_{\{x_1,\ldots,x_n\}}(x) + a_2^2 MEI_{\{x_1,\ldots,x_n\}}(x)^2$$

As explained in *Shape Outlier Detection and Visualization for Functional Data: the Outliergram* by Ana Arribas-Gil & Juan Romo [2], the principle of the Outliergram algorithm is to use the relation between MBD and MEI as introduced in the previous section. It has been shown that all the (MEI, MBD) points lie below a parabola and that the closest points to the parabola correspond to curves with typical shape, whereas the most distant ones represent outlying curves in terms of shape. Therefore, each curve is projected onto a plane defined by the axis (MEI, MBD) and the curves whose projection is under the threshold defined by $mb_i \leq P_i - Q_{d3} - 1.5 IQR_d$ (see Algorithm 1 for more details) are defined as outliers. The value of 1.5 is the default value but it is possible to modify it in two different ways. On the one hand, the user can directly choose this inflation value to rise or get down the threshold. On the other hand, the user can choose the expected number of shape outliers in the sample and an additional step in the algorithm computes the corresponding factor that multiplies IQR_d. This second version is called Adjusted Outliergram.

Nonetheless, this whole reasoning might fail with curves that lie above or below the majority of the curves in the sample, that is, with MEI values close to 0 or 1. Indeed, for such curves the modified band depth will always be low, since they are surrounded by very few curves, independently of the fact that they might present an atypical shape or not. However, if the curve presented a typical shape and and is now shifted vertically towards the center of the sample its MBD in the new location should increase (as MEI increases or decreases). On the other hand, if the curve's shape was atypical, even when placed in the center of the sample, its MBD would remain low. That motivates the addition of

```
Data: A set of velocity curves $x_1, ..., x_n$
Result: A set of index corresponding to shape outliers
for $i$ in $[[1,n]]$ do
    Compute $mb_i = MBD(x_i)$
    Compute $me_i = MEI(x_i)$
    Compute $P_i = a_0 + a_1 me_i + n^2 a_2 me_i$
    Compute $d_i = P_i - me_i$
end
Compute third quartile and inter-quartile range of the sample $d_1, ..., d_n$,
  $Q_{d3}$ and $IQR_d$
for $i$ in $[[1,n]]$ do
    if $mb_i \le P_i - Q_{d3} - 1.5IQR_d$ then
        SO $\leftarrow SO \cup \{i\}$
    end
end
for $i$ in $[[1,n]] \backslash SO$ do
    if $\exists t \in I$ s.t. $x_i(t) < min_{j \neq i} x_j(t)$ then
        Define $\widetilde{x}_i(t) = x_i(t) - min_t\{x_i(t) - min_{j \neq i} x_j(t)\}$
    end
    if $\exists t \in I$ s.t. $x_i(t) > max_{j \neq i} x_j(t)$ then
        Define $\widetilde{x}_i(t) = x_i(t) - max_t\{x_i(t) - max_{j \neq i} x_j(t)\}$
    end
    Compute $\widetilde{mb}_i$, $\widetilde{me}_i$ and $\widetilde{P}_i$
    if $\widetilde{mb}_i \le \widetilde{P}_i - Q_{d3} - 1.5IQR_d$ then
        SO $\leftarrow SO \cup \{i\}$
    end
end
```

Algorithm 1: Outliergram algorithm

a second step in the shape outlier detection procedure in which the more extreme curves are vertically shifted towards the center of the sample one by one. They would be considered shape outliers if the new (MBD,MEI) point lies in the outlying region previously determined.

To detect magnitude outliers, an extension of the classical boxplot is used: the functional boxplot. This time, the central region is defined as:

$$C_{0.5} = \{y : \forall\, t \in I,\ \exists\,(q,r) \in (1,..,\left\lfloor \frac{n}{2} \right\rfloor)^2 \ s.t.\ y_{\lfloor q \rfloor}(t) \le y(t) \le y_{\lfloor r \rfloor}(t)\}$$

where $y_{\lfloor i \rfloor}$ is the curve with the i^{th} largest MBD

```
Data: A set of velocity curves $x_1,...,x_n$ to analyse and a set of known
      curves with normal shapes $y_1,...,y_n$
Result: The set of curves to analyse that seem abnormal
$X \leftarrow \{x_1,...,x_n\} \cup \{y_1,...,y_n\}$
$O \leftarrow Outliergram(X) \cup FunctionalBoxplot(X)$
$O \leftarrow O \cap \{x_1,...,x_n\}$
```

Algorithm 2: Outlier curves detection algorithm

This 50% central region is the analogue of the inter-quartile range (IQR) and gives us a useful indication of the spread of the central 50% of the curves. This region is not affected by outliers or extreme values, therefore it can be used to detect outliers. To do so, the 1.5 times IQR empirical outlier criterion is extended to the functional boxplot. The envelope of the 50% central region is inflated by 1.5 times the range of the 50% central region. Any curves outside this inflated envelope will be considered as a potential outlier.

The final outlier curves detection algorithm is a simple combination of the two previous algorithms. Every curve detected as a shape outlier by the outliergram or as a magnitude outlier by the functional boxplot will be considered a potential outlier. The following part will explain the classification of these curves into either:

- the set of normal curves if the curve is detected as an outlier but is only an extreme value among the curves that have an usual shape (e.g the minimum and maximum acceptable falling time curves)
- one of the types of well-known problems that happen on control clusters
- the set of abnormal curves that do not correspond to any well-known problem; in this case a further technical analysis is needed

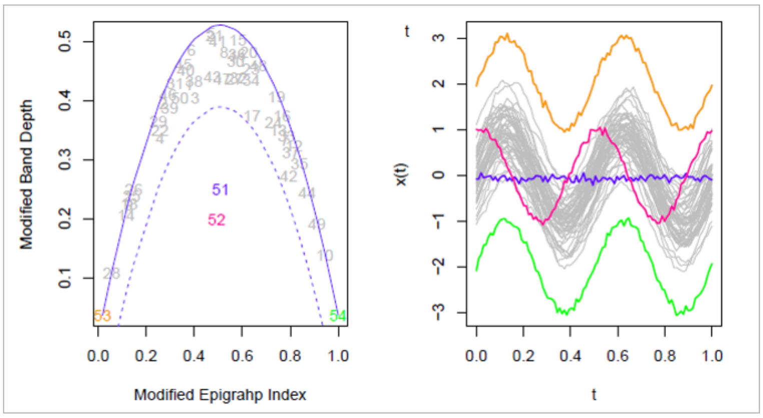


Figure 3.6: Outliergram and functional boxplot applied to generated data

The Figure 3.6 is the plot given by the final algorithm using the generated functional dataset on the right. Both pink and blue curves are detected as shape outliers since they appear under the dotted parabola on the left while the two other colored curves are detected as magnitude outliers but not as shape ones.

3.2.2 Functional K-nearest-neighbors algorithm

Standard KNN algorithm

This part refers to *k-Nearest-Neighbors Classifiers* by Padraig Cunningham and Sarah Jane Delany [4].

The K-Nearest-Neighbors (KNN) algorithm is a supervised learning algorithm. It is a non-parametric classification method in which a new observation is classified in the majority class of the input's neighborhood among the training sample.
To determine this neighborhood, a metric is required. In the case of functional data, the most common choice is the L^2 distance defined as:

$$d_2(f,g) = ||f-g||^2_{L^2}$$

An issue with this version of the KNN algorithm is that it does not take into account the distance between the outlier and each of the curves among the K nearest neighbors. This could lead to misclassifications, mostly in cases of rare problems that do not have multiple representatives in the training set. One of the solution is to weight each of the neighbors depending on the distance to the curve to classify when computing the majority vote.

```
Data: A curve $x$ to classify and a set of known curves $y_1, ..., y_n$ and their
        associated classes $C_1, ..., C_n$
Result: A given class for $x$
$D \leftarrow (d_2(x, y_i))_{i=1,...,n}$
$KNN \leftarrow argmin_K(D)$
$Class \leftarrow majority_{\{j \in KNN\}}(C_j)$
```

Algorithm 3: Functional KNN algorithm

Weighted KNN algorithm

Before introducing the disparities between the two algorithms, let us introduce a fundamental notion: the kernel function.

Definition. *A kernel function is a function $f : \Re \longrightarrow \Re_+$ such that:*

- $\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x = 1$
- $f(d) \geq 0 \quad \forall d \in \Re$
- $\max_{d \in \Re} f(d) = f(0)$
- *$f(d)$ descents monotonically for $d \longrightarrow \pm\infty$*

Examples. *The following functions are such functions:*

- *Rectangular kernel: $R : d \longrightarrow \frac{1}{2}\mathbb{1}_{|d|\leq 1}$*
- *Biweight kernel: $BW : d \longrightarrow \frac{15}{16}(1-d^2)^2\ \mathbb{1}_{|d|\leq 1}$*
- *Gauss kernel: $GK : d \longrightarrow \frac{1}{\sqrt{2\pi}}\exp(-\frac{d^2}{2})$*

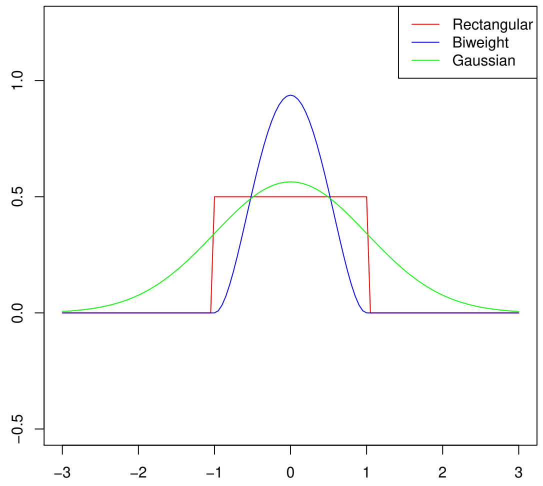


Figure 3.7: Kernel functions

The biweight kernel function is both smooth enough to take into account every neighbor and with a sufficient gradient to differentiate the neighbors on the basis of their distance to the curve to classify. See *Algorithme des k plus proches voisins pondérés et application en diagnostic* by Eve Mathieu-Dupas [8] for more details. The next step is to standardize the neighbors' distances so that they all fit in $[-1,1]$. To do so, once the selection of the K nearest neighbors done, each of their distance is standardized with the $(k+1)^{th}$ neighbor, i.e:

$$D(x, y^{(i)}) = \frac{d(x, y^{(i)})}{d(x, y^{(k+1)})} \text{ for } i = 1, ..., k$$

In particular, now every standardized distance fits in the interval $[0,1]$. Note that in the algorithm implementation, a constant $\epsilon > 0$ is added to $d(x, y^{(k+1)})$ to avoid null weights.

Once this computation done for the whole neighborhood, the new case x is given the class C of maximum weight in its K-neighborhood, i.e:

$$C = max_r \left(\sum_{i=1}^{K} BW(D(x, y^{(i)})) \mathbb{1}_{C^{(i)}=r} \right)$$

where BW represents the biweight kernel function.

Note that the main advantage of this weighted method is that it relies less on the choice of K than in the standard KNN method. Indeed, if K is chosen too high, then the influence of the furthest neighbors is reduced by the weights. Refering to *Weighted k-Nearest-Neighbor Techniques and Ordinal Classification* by Klaus Hechenbichler [6], the full algorithm is described in Algorithm 4.

```
Data: A curve $x$ to classify and a set of known curves $y_1, ..., y_n$ and their
      associated classes $C_1, ..., C_n$
Result: A given class for $x$
$D \leftarrow (d_2(x, y_i))_{i=1,...,n}$
$D \leftarrow sort(D)$
$D^{(i)}_{stand} \leftarrow \frac{D(x, y^{(i)})}{D(x, y^{(k+1)})} = \frac{d_2(x, y_i)}{d_2(x, y_{k+1})} \quad , i = 1, ..., k$
$w^{(i)} \leftarrow BW(D^{(i)}_{stand})$
$Class \leftarrow argmax_r \left( \sum_{i=1}^{k} w^{(i)} \mathbb{1}_{C^{(i)}=r} \right)$
```

Algorithm 4: Functional weighted KNN algorithm

3.2.3 Functional K-means algorithm

This part refers to *K-means alignment for curve clustering* by Laura Maria Sangalli, Piecesare Secchi, Simone Vantini & Valeria Vitelli [10].

K-means algorithm is a clustering algorithm that aims to partition n observations into k clusters. The given clusters minimize the within-cluster variances. The functional version of K-means is alike in all respects to the standard one except for the chosen metric. In what follows, the considered metric is the L^2 distance.

In the corresponding part, the R package *fdakma* [10] is used. Note that it will be used without any alignment.

3.3 Obtained results

In the following part, the algorithms have been tested on a subset of the total data that concerns nuclear power plants that belong to a specific power level for which the problems encountered with control clusters are well-known. It is important to note that the final algorithm needs to be used in this way. Indeed, some disparities may occur from one power level to another. Therefore, a curves library will be created independently for each nuclear power level.

```
Data: A functional dataset $D = \{x_1, ..., x_n\}$, a number of desired clusters k
Result: k clusters $C_1, ..., C_k$
$t \leftarrow 0$
Randomly initialize k centroids $\mu_1^t, ..., \mu_k^t$ among $x_1, ..., x_n$
repeat
    $t \leftarrow t + 1$
    $C_j \leftarrow \emptyset$ for all $j = 1, ..., k$
    for $x_j \in D$ do
        $j^* \leftarrow arg\min_i d(x_j, \mu_i^{t-1})$
        $C_{j^*} \leftarrow C_{j^*} \cup \{x_j\}$
    end
    for $i \in [\![1, k]\!]$ do
        $\mu_i^t \leftarrow \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$
    end
until $\sum_{i=1}^k d(\mu_i^t, \mu_i^{t-1}) \leq \epsilon$;
```

Algorithm 5: Functional K-means algorithm

3.3.1 Curves library creation

Figure 3.8 shows the whole dataset that will be used in this result part. Some unusually shaped curves are visible to the naked eye. As explained before, the aim here is to organize these curves into shape clusters that reflect specific control cluster issues. To do so, two steps are needed. First, a functional k-means algorithm is used in order to obtain shapes clusters. Then, these obtained clusters are refined or regrouped by a field expert.

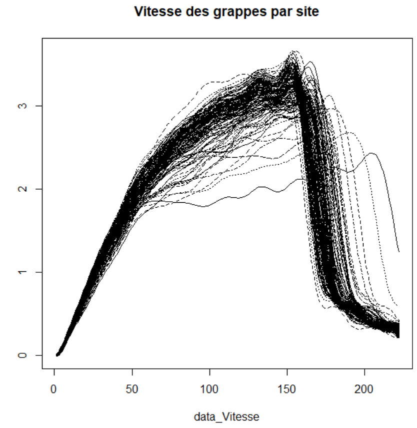
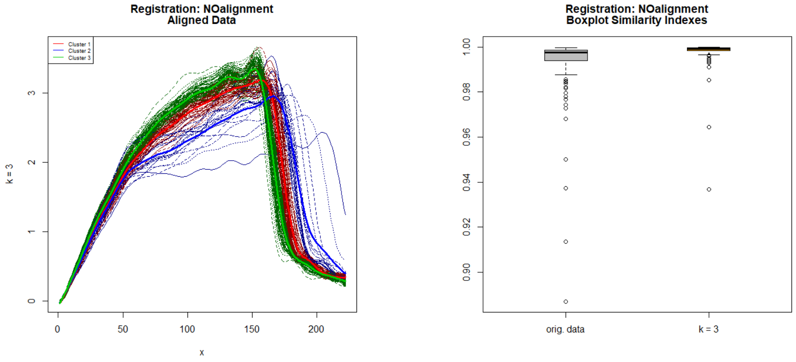


Figure 3.8: Velocity curves before clustering

The objective is to increase clusters cohesion, i.e similarity within groups, while not having more clusters than there are types of issues. Since there are at least normal shape curves and curves of control rods that fall too quickly or too slowly, the minimum number of considered clusters has to be 3. In what follows, the R package *fdakma* [10] is used. This library contains a k-means function that can be used conventionally or with an alignment feature that modifies the x coordinate for each curve in order to align them. In the case of this study, since the x coordinate is a time with a well-defined zero time and since the interest is to detect slow control rods, this feature will not be used.



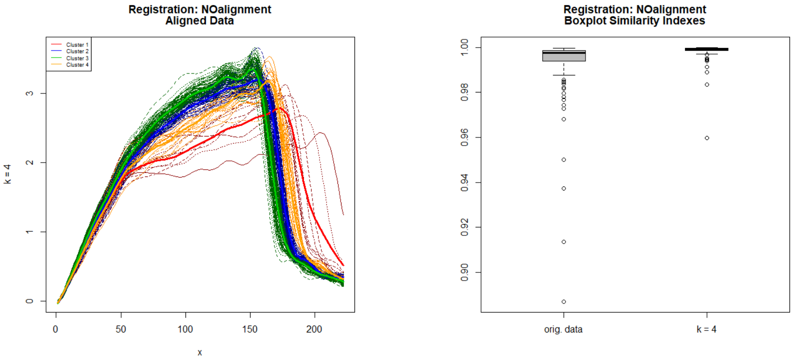
(a) Obtained clusters for k=3

(b) Similarity within clusters boxplots before and after clustering

Figure 3.9: K-means output for k=3

Figure 3.9 shows that most curves have their within cluster similarity indexes almost reaching 1 after clustering. However, the curves that correspond to slow control rods belong to the second cluster, a cluster that contains as well curves that have an absolutely usual shape. Therefore, a supplementary cluster seems to be needed.

Figure 3.10 is the output of the functional k-means algorithm with 4 clusters. Once again, the boxplot after clustering is satisfactory overall. Moreover, this time, a whole cluster, the first one, is dedicated to slow control clusters. After consultation with a field expert, it has been agreed that clusters 1 and 3 are good representatives of both slow and fast control clusters with the exception of a few adjustments and that clusters 2 and 4 are as well good representatives of usual control rods velocity curves. Therefore, these last two clusters are regrouped.



(a) Obtained clusters for k=4

(b) Similarity within clusters boxplots before and after clustering

Figure 3.10: K-means output for k=4

Note that this part of the study needs to be done for each power level and that the number of final clusters is not fixed.

3.3.2 Outlier curves detection

Once the curves library creation completed, the aim is to be able to detect unsual curves among new datasets. The idea is to spot both control panels whose velocity curves have odd-looking patterns and control panels reaching too high or too low velocities. To do so, two different algorithms are combined: the Outliergram and the Functional Boxplot. The first one is aiming to detect shape outliers while the second one is made to spot magnitude outliers.

To show this combination of algorithms' outcome, the following tests have been carried out with the same dataset as in the previous section. In this way, all the control clusters are comparable since they have the same structure and the same physical properties. In this dataset in particular, some of the curves are expected to be abnormal since the free-fall time of the corresponding control clusters is already unusual. Therefore, these particular curves have to get detected. There is no worry about detecting some additional curves since the following classification part is simultaneously a way of giving a first diagnosis and a way of reclassifying the normal curves detected as outliers.

Figure 3.11 represents the output of the detection algorithm without outliergram adjustment. On the right hand side is the projection of each curve onto the plane (MEI,MBD). The theoretical parabola fits far less well than with a generated dataset. It is quite logical since these are true acquisition that fluctuate de-

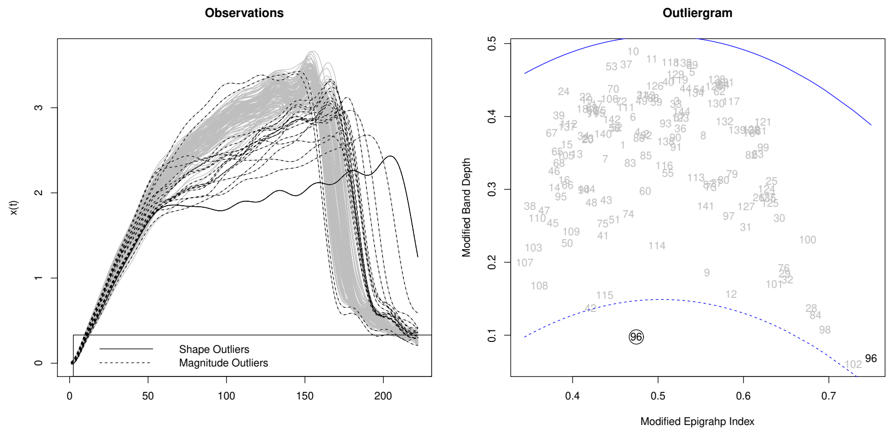


Figure 3.11: Outliergram and functional boxplot applied to control cluster velocity curves

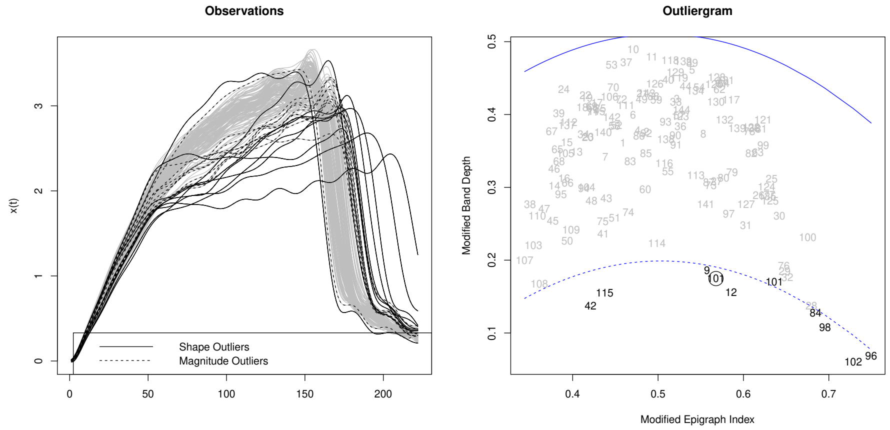


Figure 3.12: Adjusted outliergram (5%) and functional boxplot applied to control cluster velocity curves

pending on lots of physical phenomena. Observe that a circled number means that this curve has been vertically shifted towards the center of the sample to determine whether it is a shape outlier or not. On a positive note, every curve that looks abnormal is detected either by the outliergram or by the functional boxplot. Nonetheless, the outliergram alone seems not to be sufficient since it only detects few curves and that some other shape outliers are visible for the trained eye.

However, both of the algorithm's parts are needed in order to be efficient since each abnormal curve is both a shape and a magnitude outlier (the area under the curve being the same for each curve). In this particular case, the Adjusted

Outliergram seems to be more effective. Indeed, in this case, a little less than 5% of shape outlier curves is expected. Therefore, the adjusted algorithm is given as an input a value of 5% to be sure not to miss any of these curves. This time, the curves with clear unusual shapes are detected by the outliergram alone and others are detected only by the functional boxplot. Moreover, all the expected abnormal curves are among the algorithm outputs.

Nonetheless, the final algorithm lets the user chose between the Standard and the Adjusted Outliergram since there are still cases in which the standard one gives excellent results.

These two tests shows the effectiveness of the chosen outliers detection algorithms. Note that in order to detect outliers among new curves that are not part of the known curves library, the outliergram and the functional boxplot are applied to an artificial functional dataset composed of the new curves to examine and the known curves that are known to have usual shapes. Then, the returned outliers are the curves that both belong to the new dataset and are detected as shape or magnitude outliers by the algorithm applied to the artificial dataset.
The aim of this whole project is to produce a tool that gives a first diagnosis to any control cluster for which a velocity curve has been recorded. At this point, any new curve can be classified as usual or unusual. This classification is enough for usual shape curves but a more precise diagnosis needs to be done for outliers; this is the topic of the following section.

3.3.3 Outlier classification

Until now, the diagnosis of a problem happening on a control cluster has been done on a case-by-case basis. After an inconclusive test had been carried out, the maintenance engineer took a look at the corresponding curve and was either able to recognize a well-known problem or had to check for new conditions that may have caused the problem. Therefore, there is a good knowledge of the various problems involving control clusters. The idea is to use this professional knowledge to automate the diagnosis process, that is to use a supervised learning algorithm.

Experience shows that two control rods which curves look alike are likely to share the same issue. Therefore, the choice fell on using a weighted K-nearest neighbors with L^2 distance as metric and the known curves library as training set. The weighted dimension of the algorithm is there to classify new curves in classes of issues that have a limited presence in the training set.

Unlike in the case of the standard KNN algorithm, this choice here is not decisive. Indeed, while having too few neighbors may still skew the outcome, the risks related to having too many neighbors are cancelled by the weighted dimension of the algorithm. Anyway, a minimum number of neighbors to be considered needs to be determined. To do so, a 10 fold cross-validation is done using as dataset an other well-known nuclear power level for which the library creation has been done in the same way. For each value of k, the micro-averaged F1-score is computed. Is is a generalization of the F1-score for multi-class prediction. In the case of multi-class, it can be proved that it is equivalent to the micro-accuracy and computed as:

$$F1_{micro} = Acc_{micro} = \frac{TP}{TP + FP}$$

, where True Positives correspond to values that are correctly predicted and False Positives to prediction errors.

The obtained results are shown on Figure 3.13. An elbow can be observed for $k = 10$ and this will be the chosen number of considered neighbors for the final algorithm. Note that the obtained performance does not seem to decrease with a large number of neighbors while it would have decreased a lot with the standard KNN method. This shows the interest of using weights and kernel functions with an important gradient around zero.

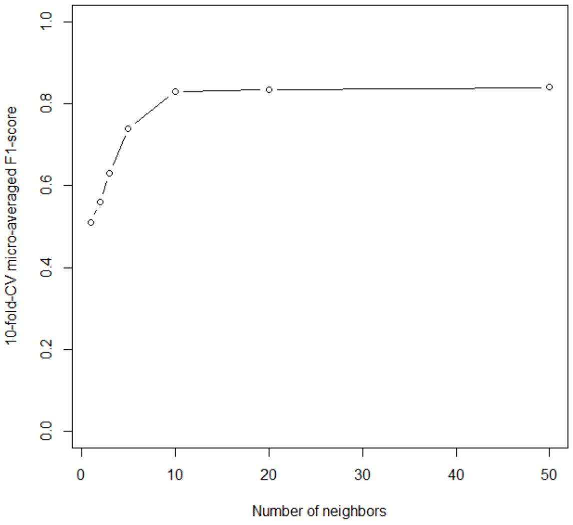


Figure 3.13: Number of neighbors choice cross-validation

Now that the weighted KNN is ready for use, let us test the whole classification algorithm on the dataset that has been used for the curves library creation part. To that end, the dataset is randomly divided into a training and a test sets whose sizes are respectively 80% and 20% of the initial dataset's size. The training set is used as the curves library while the test set represents new curves that need to be classified.

The training set contains 116 curves divided into three classes: usual shape curves, issue1 and issue2. The description of the issues are hidden to maintain confidentiality. These curves are plotted with their respective classes on Figure 3.14.

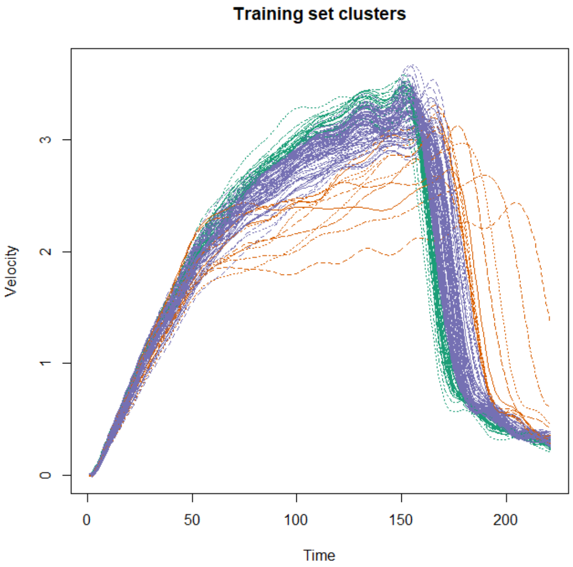


Figure 3.14: Training set clusters. Blue curves correspond to usual shape curves while green and red ones correspond respectively to Issue1 and Issue2

Then, the 29 remaining curves are given as inputs to the detection and classification algorithm. To compare the obtained results with the real classes, Figure 3.1 represents the corresponding confusion matrix.

True / Predicted	Usual	Issue1	Issue2
Usual	23	1	0
Issue1	0	1	0
Issue2	0	0	4

Table 3.1: Confusion matrix for the test set

Only one abnormal velocity curve seems to be misclassified. This can be explain by the fact that the Issue1 class (in green on Figure 3.14) contains curves that represent control rods that are a bit too fast but that do not represent any danger. Therefore, their shapes are very similar with the usual ones. Note that the outlier detection algorithm returns 7 outliers and that one of them is successfully reclassified as usual by the weighted KNN algorithm. This highlights the importance of combining these two algorithms consecutively.

At this point the algorithm is functional to detect existing issues. Nonetheless, two main questions remain:

- How to deal with new issues that do not appear in the curves library ?
- How to ensure that the classification is correct ?

First of all if a new curves correspond to an unknown issue, there is at this point no means to detect it: the curve will be classified into one of the existing classes in the curves library. Then, if nothing else but a single class is returned by the algorithm, the user cannot detect whether the class has been chosen unanimously or if there were two or more classes competing for it. Therefore, some supplementary information needs to be returned along with the output class.

First of all, a graphical output is given to the user. It contains the test set clusters with the same colors as in Figure 3.14. It allows the user to check the accuracy of the classification by eye. This graph is returned along with a PCA plot. To perform this PCA, the first two principal components are computed using as input data the training set. Then, each test set curve is projected on the plane defined by $(PC1, PC2)$. Figures 3.15 and 3.16 show these outputs.

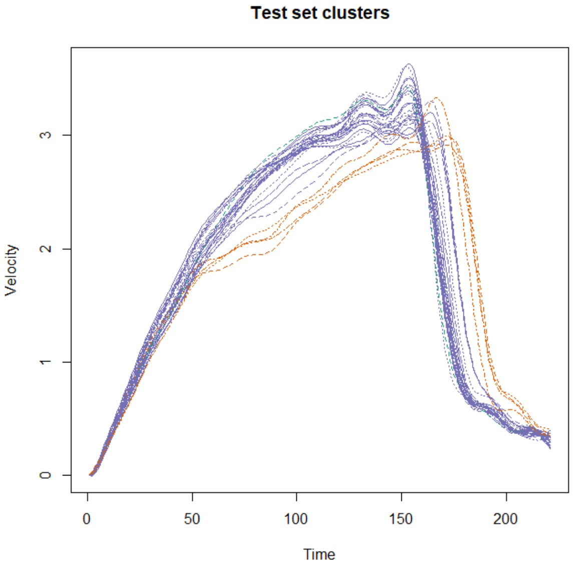


Figure 3.15: Test set clusters. Blue curves correspond to usual shape curves while green and red ones correspond respectively to Issue1 and Issue2

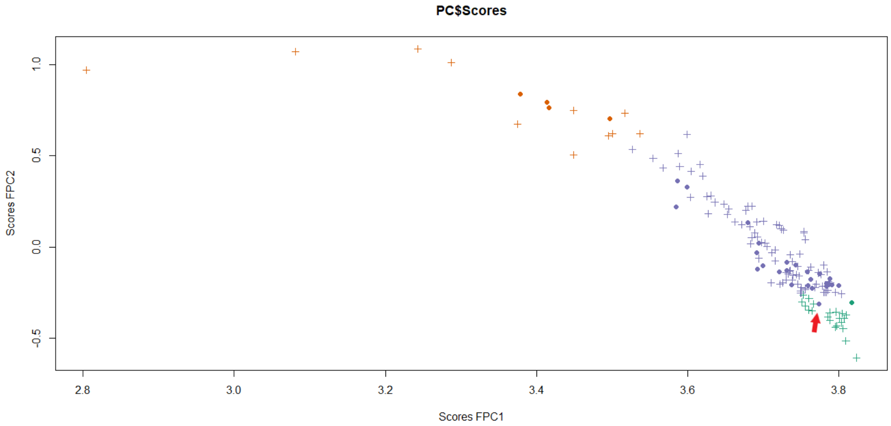


Figure 3.16: Projection on the first two principal components computed on the training set. The colors are the same as the ones used to represent clusters. Crosses are the projection of the training set while dots are the test set's ones

The PCA part is in this example very useful to detect that the misclassified abnormal curve is in fact very close to curves that belong to the Issue1 class. The projection of this specific curve is indeed a purple dot surrounded by green crosses. Nonetheless, a new curve that would correspond to an unknown issue could not appear as unusual on the principal components projection since they are computed on the curves forming the library. Therefore, the last output returned by the algorithm is the distance between the input curve and its closest neighbor among the curves library. This additional information gives an indication of the functioning of the final classification algorithm. If this value is about average, the classification is probably correct while if this value is higher than usual, a closer look on this specific curve needs to be taken. If this curve happens to correspond to a new problem, it is added to the known curves library to detect future appearances of this phenomenon.

3.4 Discussion

The work carried out shows that it is possible to automatically detect atypical falling time curves in a homogeneous and reproducible manner. It is also possible to propose a first diagnosis on the origin of the atypical character and thus facilitate the analysis of atypical curves. Finally, the presentation of the results in a graphical form allows a non-statistician user to easily detect a curve corresponding to a new problem.

Overall, the feasibility of the approach is demonstrated and the proposed method meets the expressed need.

However, the databases still need to be enriched with more curves for each level, ideally all the curves recorded over a period of 5 years, i.e. about 40,000 curves on the scale of the 56 units operated by EDF. This will enable clusters to be more representative of all the phenomena encountered and also more robust. On this basis, the automatic and systematic analysis of the new curves can be implemented.

Chapter 4

ARE valves behaviour inventory

4.1 Introduction

4.1.1 Background

The water level in the steam generators is an important parameter for the safety of nuclear reactors: it guarantees the availability of a sufficient quantity of water to cool the reactor in the event of an incident. To ensure that it remains within the expected ranges, it is regulated by an automaton that uses two levers that play on the flow of water entering the steam generators: the rotation speed of the feed turbopumps and the opening of the "ARE valves". When this regulation is not sufficient to keep the expected water level in the steam generators, the reactor protection mechanisms lead to an automatic shutdown in less than 2 seconds by triggering the fall of the control clusters.

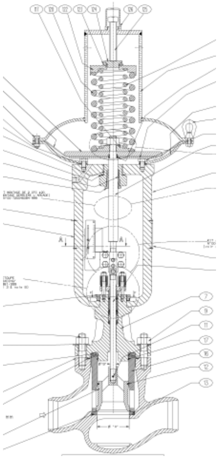


Figure 4.1: ARE valve

The purpose of the study is therefore to develop indicators that could be used to monitor the correct operation of this regulation and thus avoid automatic shutdowns by detecting drifts before failures.

4.1.2 Methodology

Feedback shows a greater frequency of this type of shutdown on the 12 units of the P'4 stage. It is thus on these units that the study will focus more precisely on the behavior of the valves.

The speed variations of the food turbopumps are made in a few minutes when a few seconds are enough to change the opening of an ARE valve.

The first step of the study consists in finding a method to filter the effects of the speed variations of the feed turbopumps on the behavior of the taps and then to propose indicators to characterize it.

Then, these indicators are calculated over a period of 20 years for the 12 slices studied. The analysis of their distribution then made it possible to specify the intervals of the expected values in order to identify atypical behaviours more easily.

Finally, the evolution of these indicators over the month(s) preceding the various automatic shutdowns that have occurred (or almost occurred) at the power plants was analysed in order to verify their relevance.

4.1.3 Data extraction

A way of examining the behaviour of a specific ARE valve is to have a look at the evolution of the flow rate in the corresponding pipe. There exists an internal tool called Curiosity which is an API containing a huge database and several widgets that can be used to extract and manipulate the data. Curiosity is a tool developed by EdF R&D to facilitate the analysis of nuclear park data. In this study, it will be used to access the instantaneous flow rates of the ARE circuit. The different calculations (running average, difference between raw values and running average, indicators) are then calculated in python scripts developed specifically for this study.

The goal here is to examine the behaviour of the mechanism involved in short-time regulations of the opening and the closing of the valve. Therefore, the flow rate fluctuations caused by the turbopump's rotation speed variations need not to be taken into account. The period of these regulations is of a few minutes while the valve's period is of a few seconds. To deal with this issue, let us introduce the moving average as follows.

Definition. *Moving average*
Let $(x_i, t_i)_{i\in[1,n]}$ *be the measurements and the corresponding times of measurement. Then, we define the simple moving average over t in* t_i *as:*

$$\bar{x}_{i,t} = \frac{1}{N_{i,t}} \sum_{t_j \in T_{i,t}} x_{t_j}$$

where $T_{i,t} = [t_i - \frac{t}{2}, t_i + \frac{t}{2}]$ *and* $N_{i,t} = |T_{i,t}|$.

By extracting the flow rate, computing the moving average over 2 minutes in each point and subtracting the moving average to the raw data, a good approximation of the fluctuations of the flow rate directly caused by the opening and the closing of the study's ARE valve (see Mean difference graph on Figure 4.2) is obtained. Flow rate raw data contains in general a point every 2 seconds with some missing values from time to time and missing data over a longer time when the power station is shut down. In order to compute easily the moving average during operating periods, a linear resampling function with a duration time of 2 seconds is used. Its principle is simply to interpolate linearly between the two closest values when a point is missing.

In order to be able to compare two different operation periods effectively, interesting indicators from the mean difference graph can be extracted. The two statistics that are both easy enough to look at and that enable the characterization of the valve's operations are the frequency and the amplitude of the instantaneous flow rate. On a time period T, let us use the two following indicators:

- the standard deviation, that gives an estimation of the mean amplitude over T
- the number of zero crossings - or pseudo-frequency - that gives an estimation of the mean frequency over T

4.2 Statistics monitoring over the country's nuclear power stations

The first part of the mission entails doing an assessment of the evolution of both the indicators on a large time scale and on every nuclear power plant that have had issues with this kind of valves. The indicators have finally been computed on the whole P'4 level, for a total of 48 sensors, on a time scale from January 2010 to December 2019 with a 24 hour extraction per month. This choice of time

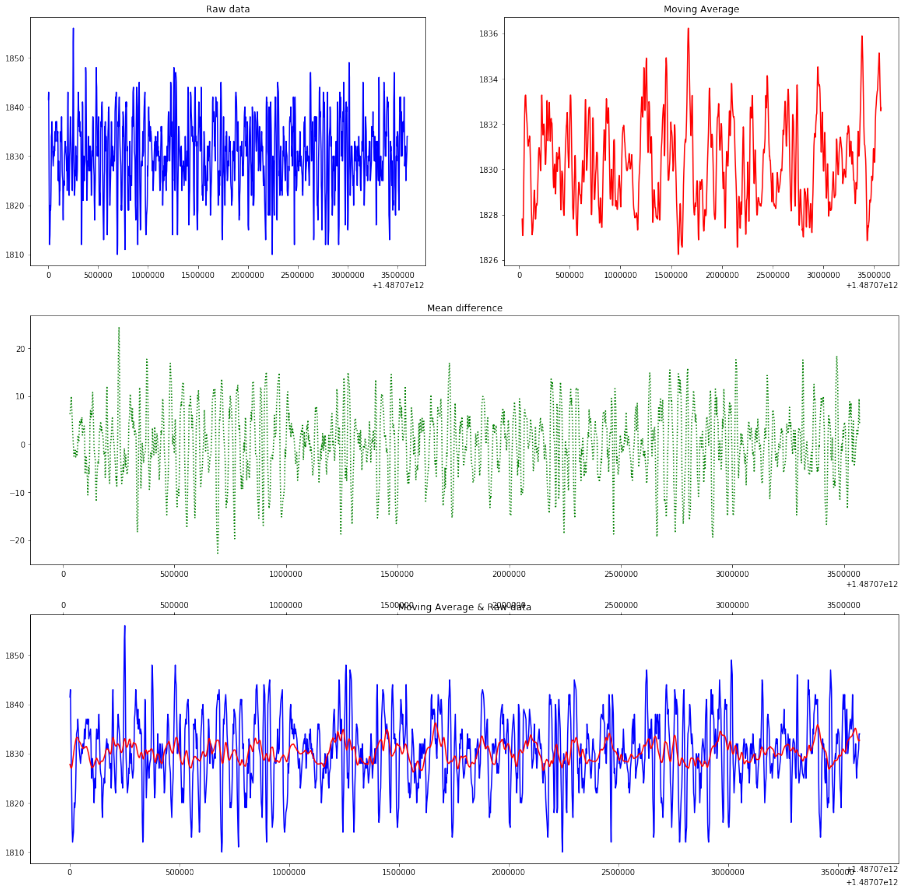


Figure 4.2: One hour of raw data, moving average & mean difference

scale is representative enough of the variations that both indicators may have while not being too CPU-intensive. Of course, since the nuclear power station are regularly under refuelling, there are some missing values which correspond to blanks or zero-value troughs on the figure 4.3 graphs.

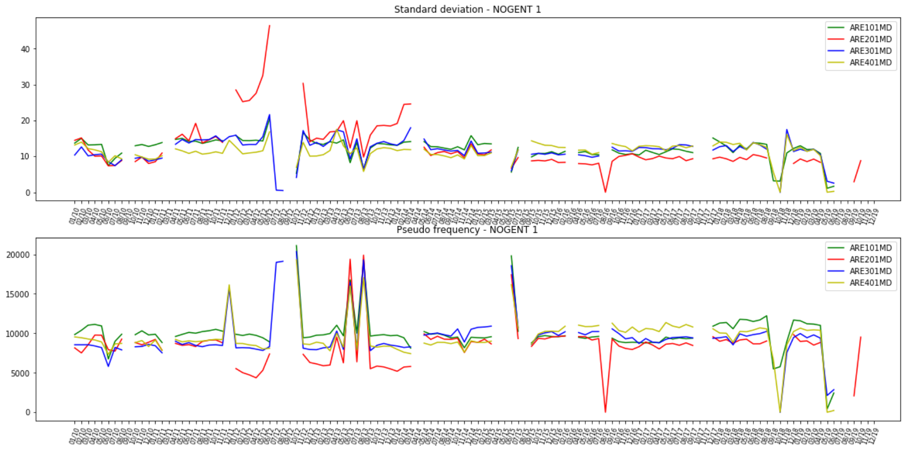


Figure 4.3: 10 years extraction of both the indicators on a nuclear station

Once the computations done for each nuclear power plant, two datasets are created: one containing the pseudo-frequency values and another one containing the standard deviations. Then, a filter is applied, removing both the pseudo-frequency and the standard deviations whose values are close to zero since they reflect either white noise around a constant value or a shut down power station. A second filter is applied to the pseudo-frequency, removing the values over 15,000 using Nyquist-Shannon sampling theorem. At this point, the objective is to fit both data samples with appropriate distribution in order to set up confidence intervals for further data. Let us introduce two measures that can be used to determine the distributions to fit the data samples with.

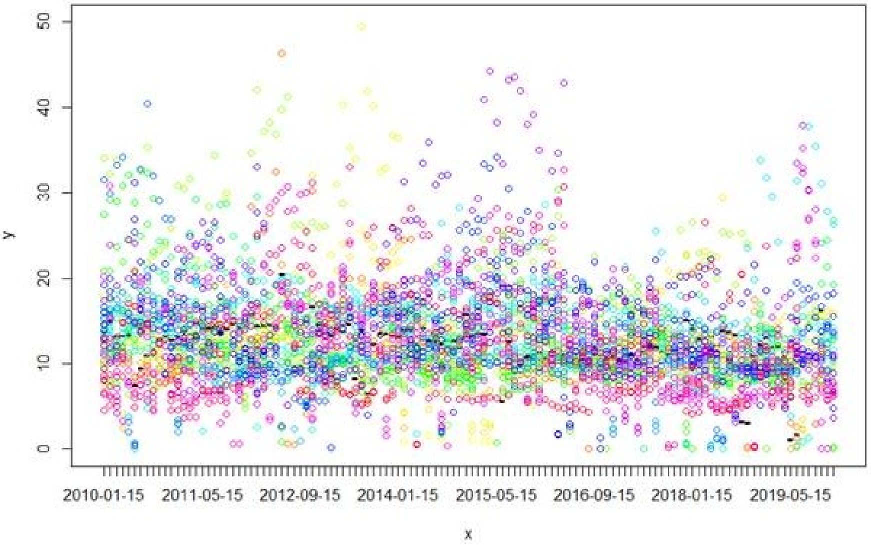


Figure 4.4: 10 years extraction of both the standard deviation on a every P'4 nuclear station

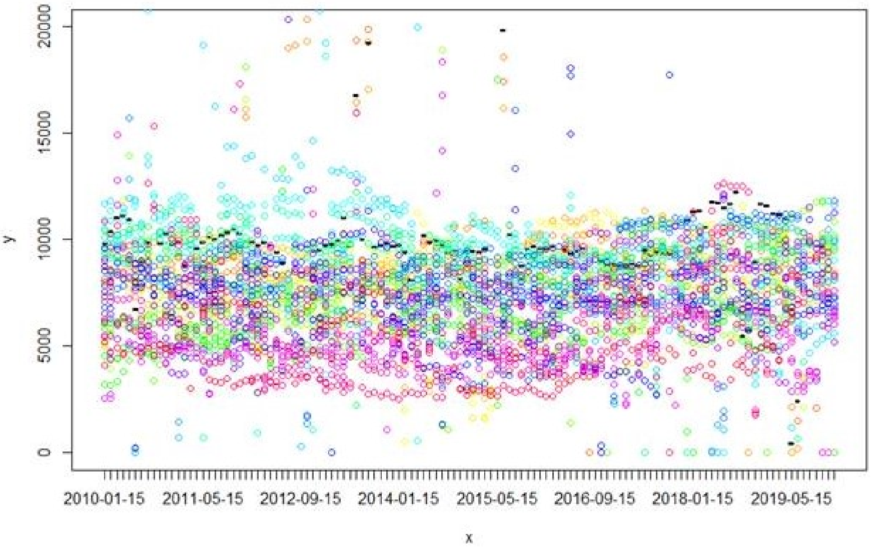


Figure 4.5: 10 years extraction of both the pseudo-frequency on a every P'4 nuclear station

Definition. *Skewness*

For univariate data $Y_1, ..., Y_N$, skewness is defined as:

$$g_1 = \frac{\sum_{i=1}^{N}(Y_i - \bar{Y})^3}{N \cdot s^3}$$

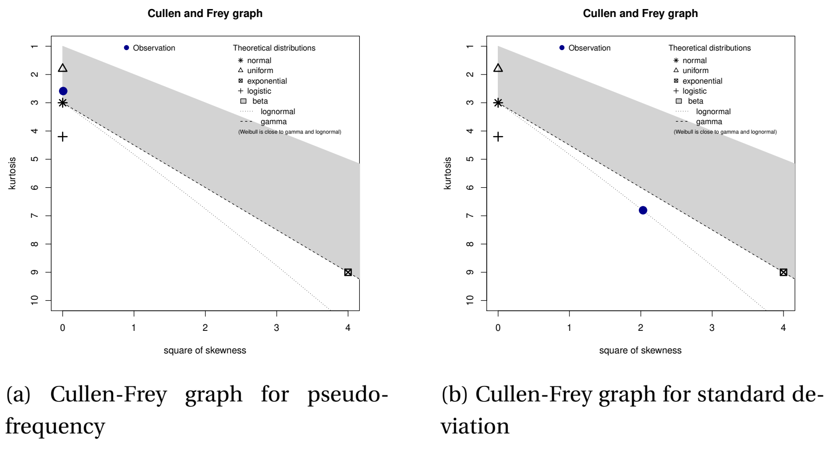
where $\bar{Y}$ is the sample mean, s the standard deviation and N the sample size. Skewness is a measure of the lack of symmetry among the sample. It is zero for a normal distribution or any other symmetric data, negative for skewed left data and positive for right skewed ones.

Definition. *Kurtosis*
For univariate data $Y_1, ..., Y_N$, kurtosis is defined as:

$$kurtosis = \frac{\sum_{i=1}^{N}(Y_i - \bar{Y})^4}{N \cdot s^4}$$

Kurtosis is a measure of whether the data are heavy-tailed or light-tailed in comparison with a normal distribution. Its value is 3 for a standard normal distribution.

A way of determining easily the best distribution to fit a data sample with is to plot the kurtosis of it against its skewness, knowing well the theoretical values of well-known distributions, on the so-called Cullen-Frey graph. Then, the density whose theoretical $(skewness, kurtosis)$ point is the closest to the data sample's one is chosen. Let us plot the Cullen-Frey graph for both pseudo-frequency and standard deviation using R package *fitdistrplus* [5].

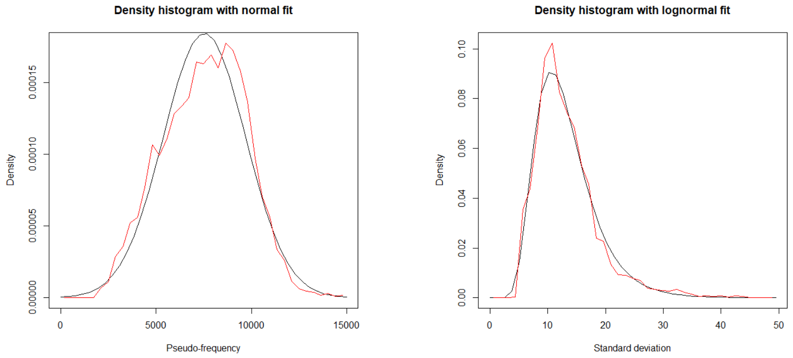


(a) Cullen-Frey graph for pseudo-frequency

(b) Cullen-Frey graph for standard deviation

Figure 4.6: Cullen-Frey graphs to determine the density to fit with

The lognormal distribution seems very well-suited for the standard deviation. For the sake on simplicity, the pseudo-frequency sample will be fitted using a gaussian distribution.



(a) Real data and gaussian fit for pseudo-frequency

(b) Real data and lognormal fit for standard deviation

Figure 4.7: Fitted models for pseudo-frequency and standard deviation

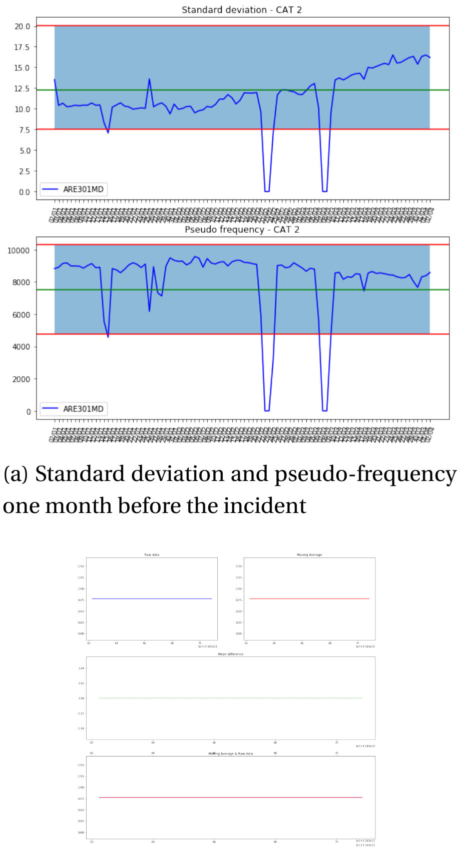
The obtained fitted models are the following:

$$\begin{aligned} P_f &\sim \mathcal{N}(7536, 2165) \\ Std &\sim Log - \mathcal{N}(2.5, 0.38) \end{aligned}$$

4.3 Evolution of the indicators before specific events

The objective here is to determine whether the statistics that have been defined for the purpose of preventing future incidents on ARE valves are relevant or not. To do so, several types of incidents that have involved ARE valves are extracted from a maintenance database. Then, the indicators are computed on each day during a substantial period preceding each incident and plotted on a graph on which are also plotted the mean and the 80% fluctuation interval of the corresponding indicator.

Figures 4.8, 4.9, 4.10 and 4.11 show different types of incidents that have happened on several P'4 ARE valves.

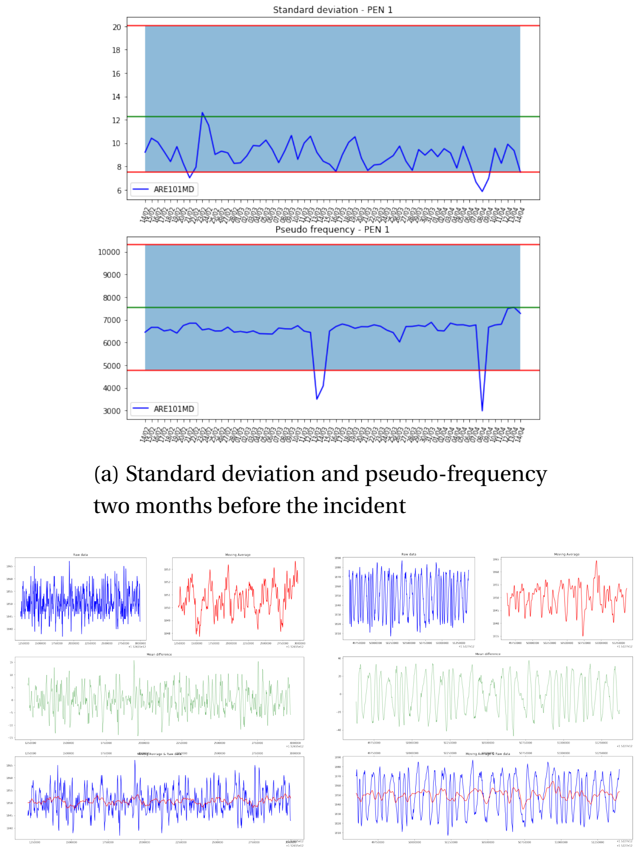


(a) Standard deviation and pseudo-frequency one month before the incident

(b) Raw data on one of the zeros that shows missing data

Figure 4.8: Slow increasing of standard deviation and decreasing of pseudo-frequency during the two weeks preceding an incident

Let us have a closer look at Figure 4.8. Since the two visible troughs are not relevant and the fact that none of the indicators leaves the blue zone, the only hint that could suggest a potential incident is the change in slope happening 2 weeks before the occurrence. Therefore, the first conclusion is that the indicators may help preventing future incident not only by looking at their value but also by looking at their gradients.



(a) Standard deviation and pseudo-frequency two months before the incident

(b) Raw data on the first day of the extraction

(c) Raw data on the day corresponding to the second trough

Figure 4.9: Two significant pseudo-frequency troughs before the incident

On Figure 4.9, the troughs seem to be early-warning signs for the upcoming incident since they are not missing values. These frequency drops do not coincide either to any standard deviation particular fluctuation. Therefore, they may be signs of losses of responsiveness on the valve's actuator. thus, the indicators and the blue acceptance zone may be enough to prevent further incidents of this type.

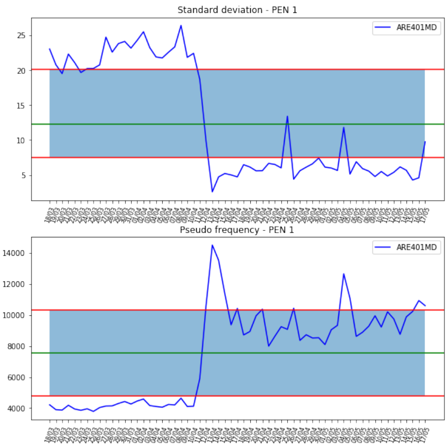


Figure 4.10: Sudden change of both standard deviation and pseudo-frequency values a month before the incident

The pattern on Figure 4.10 is rather clear. Both the indicators are relatively stable until the occurrence happening on the 10/04. Then, the circuit keeps operating despite the unusual values until the automatic switch-off. The sudden change is highly visible but would be poorly detected by the current approach since the older values are already outside the acceptance zone. Thus, a tool detecting the sudden changes regardless the pointwise values obviously needs to be included in the prevention algorithm.

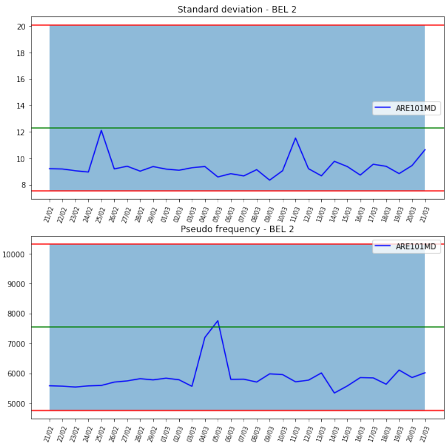


Figure 4.11: Case in which the indicators do no hint the final incident

However, sometimes the indicators do not show any sign of what could be a potential incident. This is the case on Figure 4.11.

4.4 Discussion

This study proposes two indicators to monitor the behavior of the valves contributing to the regulation of the water level on the steam generators of the P'4 power plants. It shows that it is possible to calculate them over a long period of time and over all the units concerned and thus obtain an overall view of the situation in the Park. It also shows that the evolution of these indicators seems to be linked to proven failures.

Nevertheless, before its industrial and systematic implementation of monitoring based on these indicators on French nuclear power plants, it is necessary to:

- consolidate these initial results by analyzing their evolution during normal operating transients and on all observed automatic shutdowns,
- analyze the link between the value and evolution of these indicators and the condition of the valves observed during maintenance operations.

Convinced by these first results, EDF is investigating how to finalize this study with the support of R&D with the aim of implementing a monitoring of these valves.

Chapter 5

Control cluster guides wearing estimate

5.1 Introduction

5.1.1 Background

On pressurized water reactors operated in France, the insertion of control clusters into the reactor is one of the two means used to control reactivity. These clusters are made up of 24 rods, themselves made up of a sheath containing neutrophage materials: inserting the bundles into the reactor thus makes it possible to capture neutrons that are no longer available for the nuclear reaction, which slows down the reaction. The total insertion of all the control clusters allows the shutdown of a reactor in less than 2 seconds.

A pencil sheath is a steel tube of a little less than 10 mm in diameter for a length of more than 4 m: the control clusters are thus very flexible and need to be guided during their insertion into the reactor.

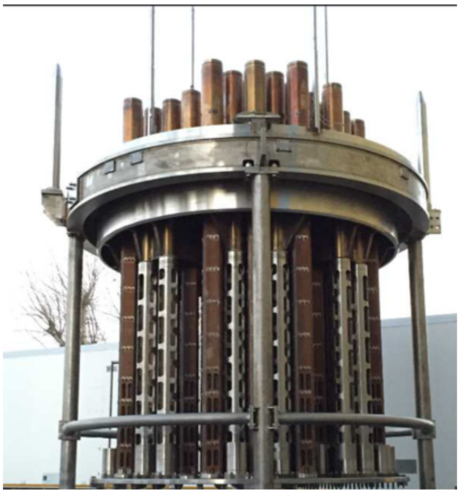
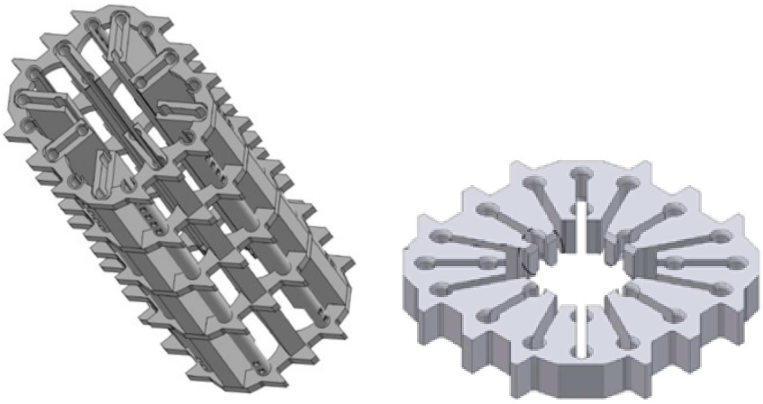


Figure 5.1: Top of a control cluster

This function is provided by a cluster guide. It is a fixed mechanical structure, consisting of a tube containing guiding cards: steel plates machined to allow the cluster's pencils to pass through.



(a) A cluster guide

(b) A guiding card

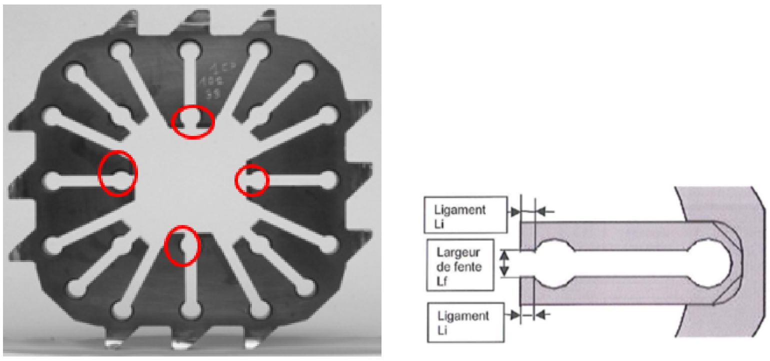
Figure 5.2: Mechanical structures guiding the control clusters during their insertion into the reactor

Due to friction between the control cluster pencil sleeves and the guide maps, the guide maps wear out. A wear limit must not be exceeded in order to guarantee the correct guidance of the control clusters in all situations. To ensure that this limit is not exceeded, wear volume measurements are taken on a regular basis. If the criteria for ensuring the correct guidance of the control clusters are not met, the cluster guide is replaced. This replacement is a relatively cumbersome operation that mobilizes scarce resources.

Therefore, before a cluster guide inspection and replacement operation, it is necessary to realistically estimate the number of cluster guides that may need to be replaced. Until now, these forecasts have been made on the basis of a linear extrapolation of the worn volumes measured during the previous inspection. Nonetheless, feedback shows that this method leads to overestimating the number of cluster guides to be replaced and to mobilizing too many resources. The purpose of this study is to propose a more realistic approach to better estimate the number of cluster leaders to be replaced and to better size the resources to be mobilized.

5.1.2 Worn volume measurements

When a cluster guide is inspected, three measures are made on each of its guiding cards' main bores. These measurements are called the gap width L_f and the ligament lengths L_i, $i = 1,2$ (see Figure 5.3).



(a) Main guiding card bores

(b) Gap width and ligament lengths

Figure 5.3: Measurements made when controlling the worn volume

In theory, the worn volume can be computed from L_1 and L_2. However, when the ligaments are too thin or poorly measured, L_f is used instead to avoid measurement uncertainty. The worn volume is computed as follows:

- If the guide is of type 1:
  - $VOLUse = -75.89 \cdot \frac{L_1 + L_2}{2} + 220.39$ if L_1 and L_2 have been correctly measured
  - $VOLUse = 24.302 \cdot L_f + 67.44$ otherwise

- If the guide is of type 2:
  - $VOLUse = -77.893 \cdot \frac{L_1 + L_2}{2} + 232.7$ if L_1 and L_2 have been correctly measured
  - $VOLUse = 28.648 \cdot L_f + 53.98$ otherwise

Note that L_f is not always measured, mainly in cases in which L_1 and L_2 are easily measurable.

To decide whether a guide needs to be changed or not, the criterion is the following: the guide is changed if the mean worn volume computed on the four main bores exceeds $328mm^2$ on at least five consecutive guiding cards.

5.1.3 Data cleaning

The study dataset contains 68.000 measurements of L1, L2 and Lf and the corresponding computed worn volume. It also contains the number of hours the guide spent in contact with a control cluster of type S (NbhS), a type C one (NbhC) and a type N one (NbhN). Note that a worn volume was given in the initial dataset; this value of worn volume was computed taking into account measurement uncertainty and corresponded to the maximum of the obtained interval. For that study, the volume has been recomputed using the formulas introduced earlier.

ID	**Year**	**L1**	**L2**	**Lf**	**VOLUse**	**NbhS**	**NbhC**	**NbhN**
AAA1 A01 E1 GD01	2014	2.25	2.25	6.5	90.35	62371	127166	0
BBB1 A07 E3 GD04	2017	1.99	2.8	NN	68.98	66361	0	151987
...	...	...	...	...	...	...	...	...

Table 5.1: Wear measurements dataset

The ID represents the nuclear power plant, the guide position as well as the guiding card and the main bore on which the measurement has been made.

First of all, since the study focuses on the evolution of the worn volume between two controls, only the guides whose wear has been measured at least twice are kept. Then, the ones that have been replaced after the first control are removed as well. Once this filtering done, let us plot the second worn volume against the first one in order to have a first idea of the available data.

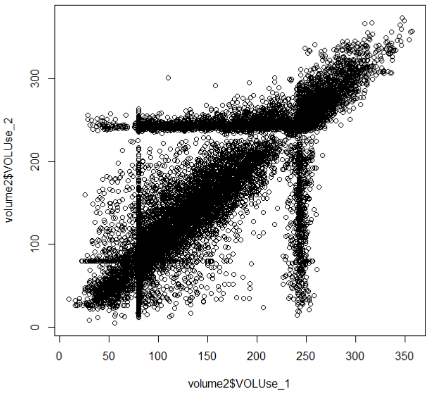


Figure 5.4: 2^{nd} measurement against the 1^{st} one

It has two be noted that the data are quite noisy. Indeed, although in mean the worn volume seems to increase between two controls, which is the expected physical result, there are still lots of points that lay under the $y = x$ curve. This is due to several things including the uncertainty of measurement, the tool change from one check to the other, etc. Then, some odd points are to notice as well: the ones forming a cross around 80mm^2 and the others forming an other cross at about 250mm^2. These points correspond to default values that are wrote down in case of difficulty during measurement (centering error, too low value of L1 or L2, etc.).

While the noise is an issue that cannot be dealt with at the risk of creating bias, the default values have to be removed from the dataset since they create bias by themselves.

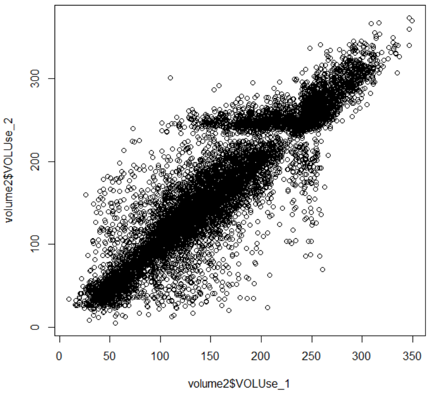


Figure 5.5: 2^{nd} measurement against the 1^{st} one after data cleaning

Then, a previous study has shown that the number of hours spent in contact with a type S control cluster has no influence on the guide's wear and that there is a significant influence of the control cluster's material on the wearing rate. Let us retrieve this result. To do so, a simple linear regression of the type $VOLUse = \alpha_S NbhS + \alpha_C NbhC + \alpha_N NbhN$ is computed.

Coef	**Estimate**	**p-value**
NbhC	$8.32 * 10^{-4}$	$< 2e^{-16}$
NbhN	$1.39 * 10^{-3}$	$< 2e^{-16}$
NbhS	$3.05 * 10^{-5}$	0.17

Table 5.2: Equivalent hour coefficients with NbhS

The result confirms that the number of hours spent in contact with a type S control cluster is not significant. Therefore, the model is reduced to $VOLUse = \alpha_C NbhC + \alpha_N NbhN$. Here, the result is the following:

Coef	**Estimate**	**p-value**
NbhC	$9.88 * 10^{-4}$	$< 2e^{-16}$
NbhN	$1.50 * 10^{-3}$	$< 2e^{-16}$

Table 5.3: Equivalent hour coefficients without NbhS

with a R^2 coefficient of 0.80 . An equivalent time $Nbh_{eq} \approx \alpha_C NbhC + \alpha_N NbhN$ can now be defined; it will be used later to define the wearing rate at the time of the first measurement.

5.2 Theoretical background

In the following part, $S := \{x_i, y_i\}_{i\in[\![1,n]\!]}$ designs the input and target values of the training set.

5.2.1 Regression tree

This part refers to *Classification and Regression Trees* by Wei-Yin Loh [7].

A regression tree is a classifier that partitions the input space into a set of rectangular regions $R_j, j = 1, ..., r$ and assigns the label l_j to all input vectors x_i falling in a region R_j. In other terms, the predicted value corresponding to x_i is:

$$\widetilde{y}_i = f(x_i) = \sum_{j=1}^{r} l_j \mathbf{1}_{x\in R_j}$$

At each step, a new node n_k is created using the feature j_k and the threshold s_k. In order to minimize the error while growing the tree, a metric called node impurity is computed and minimized at each new node creation.

Definition. *Node impurity*
We define the node impurity as the local mean squared error, i.e:

$$I(n_k) = \frac{1}{N_k} \sum_{x_i \in R_{n_k}} (y_i - l_{n_k})^2$$

with n_k the node, N_k the number of training data associated to the node, R_{n_k} the rectangular region corresponding to n_k and l_{n_k} the preliminary classification of input vectors falling within R_{n_k}.

Note that at each node, the preliminary classification l_{n_k} is computed as the majority vote for the input vectors vectors falling within the corresponding region, i.e:

$$\forall x \in R_{n_k}, l_{n_k} = f_{n_k}(x) = \frac{1}{N_k} \sum_{x_i \in R_{n_k}} y_i$$

Then, at each iteration, the new feature and threshold are chosen as:

$$(j_k, s_k) = arg \min_{j,s} I(n_k^1) + I(n_k^2)$$

where n_k^1 ans n_k^2 are the two children nodes of n_k.

Once the tree fully built, the predicted value for a new input vector is the leaf classification value where it falls following the tree's nodes. In other terms, the predicted value is the classification value corresponding to the final rectangular region the vector falls within.

5.2.2 Random forests

Random forests is an ensemble learning algorithm that builds many small regression trees in parallel in order to form a single, stronger learner by averaging the result found by each tree.

For each tree in the forest, a bootstrap sample $S^{(i)}$- i.e a subset with replacement - is selected from S. Then a modified regression tree is learnt from this new sample. The algorithm is modified is the sense that at each node of the tree, instead of examining all possible feature-spits, a subset of the features $f \subset F$ is selected randomly. The node then splits on the best feature in f rather than in F.

```
Data: A training set $S := (x_1, y_1), ..., (x_n, y_n)$, features $F$, the number of
      trees in forest $B$ and the number of features to select at each node
      $n_f$
Result: A random forest containing B regression trees
$H \leftarrow \emptyset$
for $i \in [\![1, B]\!]$ do
    $S^{(i)} \leftarrow$ A bootstrap sample from S
    $h_i \leftarrow$ RandomizedRegressionTreeLearn($S^{(i)}$, $F$, $n_f$)
    $H \leftarrow H \cup \{h_i\}$
end
```

Algorithm 6: Random forest learning algorithm

Once the learning done, the prediction for a new-coming data vector is the average of the predicted value found by each regression tree componing the random forest.

```
Data: A data vector $x$ and a built random forest $H$
Result: A predicted value $\tilde{y}$
$\tilde{y} \leftarrow 0$
for $i \in [\![1, B]\!]$ do
    $\tilde{y} \leftarrow \tilde{y}$ + RegressionTreePrediction(x, $H^{(i)}$)
end
$\tilde{y} \leftarrow \frac{\tilde{y}}{B}$
```

Algorithm 7: Random forest prediction algorithm

5.2.3 Random forest prediction intervals

While determining prediction interval is straight forward in the case of a linear regression, it is less obvious for a random forest. Refering to *Quantile Regression Forests* by Nicolai Meinshausen [9], let us introduce the main concepts for prediction interval building in this particular case.

First of all, remind that random forests approximate the conditional mean $E(Y \mid X = x)$. The conditional distribution function of Y, given $X = x$, is given by:

$$F(y \mid X = x) = P(Y \leq y \mid X = x) = E(\mathbf{1}_{Y \leq y} \mid X = x)$$

Therefore, there is a strong analogy between the random forest approximation of the conditional mean $E(Y \mid X = x)$ and the conditional distribution function of Y given $X = x$. Indeed, just as $E(Y \mid X = x)$ is approximated by a weighted mean over the observations of Y, $E(\mathbf{1}_{Y \leq y} \mid X = x)$ is approximated by the weighted mean over the observations of $\mathbf{1}_{Y \leq y}$, i.e:

$$\hat{F}(y \mid X = x) = \frac{1}{B} \sum_{i=1}^{B} \left(\frac{1}{N^i_{n_x}} \sum_{x_j \in R^i_{n_x}} 1_{\tilde{y}_j \leq y} \right)$$

where $R^i_{n_x}$ is the rectangular region of tree i whose x is part of and $N^i_{n_x}$ its cardinality.

This approximation is at the heart of the so-called **quantile regression forests algorithm**.

Therefore, when using quantile regression forest instead of a standard random forest, when growing the trees, all observations in every leaf of every tree have to be recorded while only the average is needed in the case of a standard random forest.

Then, estimating the conditional quantiles - and in this way building prediction intervals - is straightforward, indeed:

$$\hat{Q}_\alpha(x) = \inf\{y : \hat{F}(y \mid X = x) \geq \alpha\}$$

This will be used later to build prediction intervals while using random forests.

5.3 Prediction algorithms

5.3.1 Linear regression

The goal here is to provide an estimation of the worn volume difference between the first two controls. The main idea is that there is a relation between the wearing rate at the time of the first control and the wearing rate between the two controls. Therefore, the first attempt will be to build a linear model that tries to estimate the worn volume difference knowing the first control wearing rate and the time between the two controls.

To do so, let us first create a new dataset whose rows will contain all information for a single ID. Note that the features Year, NbhS, L1, L2 and Lf are left behind since they are no longer useful. Once this operation done, the new dataset looks as follows:

ID	**VOLUse1**	**VOLUse2**	**NbhC1**	**NbhN1**	**NbhC2**	**NbhN2**
AAA1 A01 E1 GD01	90.35	108.12	127166	0	0	62821
BBB1 A07 E3 GD04	68.98	81.34	0	151987	0	72491
...	...	...	...	...	...	...

Table 5.4: Wear measurements comparison dataset

Now, using the regression values as well as the worn volume difference, several new features can be created:

- $H^1_{eq} = 9.88 \cdot 10^{-4} \cdot NbhC1 + 1.5 \cdot 10^{-3} \cdot NbhN1$
- $\Delta H_{eq} = 9.88 \cdot 10^{-4} \cdot (NbhC2 - NbhC1) + 1.5 \cdot 10^{-3} \cdot (NbhN2 - NbhN1)$
- $\Delta Vol = VOLUse2 - VOLUse1$
- $\Delta Vol_{th} = \dfrac{VOLUse1}{H^1_{eq}} \cdot \Delta H_{eq}$, the theoretical difference of worn volume if the wearing rate remains the same in time

In theory, there should be a strong relation between ΔVol and ΔVol_{th}.

First attempt The first try is a simple linear model of the form $\Delta Vol = \alpha_{th} \cdot \Delta Vol_{th}$.
The obtained result shows an α_{th} coefficient of 0.20 with a very poor R^2 of 0.05. It shows that in mean, the wearing rate decreases after the first control which is the expected result but no prediction can be made with a model that inaccurate. A plot of ΔVol against ΔVol_{th} confirms this analysis.

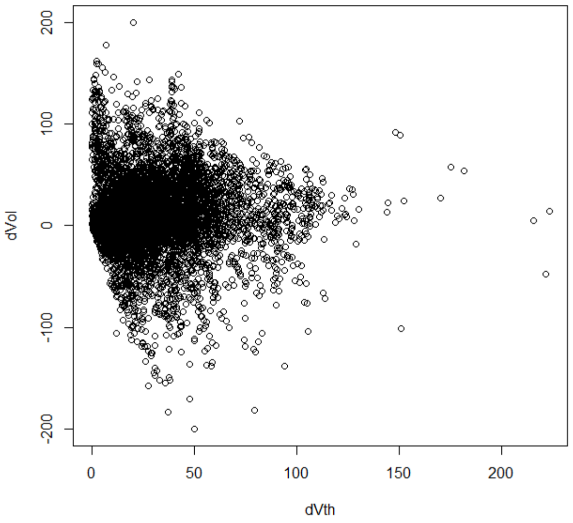


Figure 5.6: ΔVol against ΔVol_{th}

Second attempt Are the data for which ΔVol is negative located in specific nuclear power plants ? If it is the case, these data could be removed from the dataset without creating bias since their unexpected values already come from mistakes or wrong measurements that create bias. Let us have a look at the distribution of ΔVol values among each nuclear power plant (the plant names are anonymized).

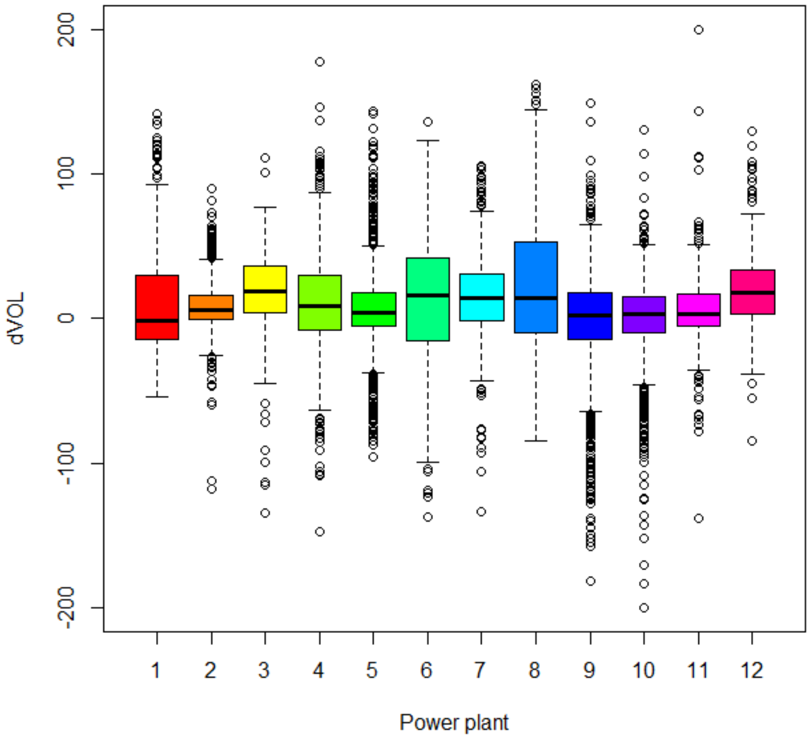


Figure 5.7: Boxplots of ΔVol for each power plant

No power plant seems to step out in term of mean but some of them show lots of highly negative values, that cannot only be due to the uncertainty of measurement. Let us remove all the data from power plants 5, 9 and 10 and try again the previous simple linear model.

This time, an α_{th} coefficient of 0.32 with a R^2 coefficient of 0.09 are obtained. These results are a bit better than the previous ones but still don't meet the needs.

Third attempt To make the linear model more efficient, some of the covariates will be added to it. Amongst the possibilities, the following features are chosen to complete the model:

- the **nuclear power plant**, since the measurement method may differ from one another as well as the working power
- the **guides' location and bore**

- the **volume at the time of the first control** and the **number of equivalent hours between the two controls**, since the wearing rate at the time of the first control might not be sufficient in itself to explain its evolution later on

This time the R^2 coefficient reaches a value of 0.19, which is again a bit better but still not enough to expect exploitable worn volumes predictions.

In conclusion, the data noise coupled with the non-linear behaviour of the wear phenomenon seems to making the choice of using a linear model ineffective. Therefore, no attempt to make proper tests with this method will be done since poor results are expected. Instead, let us move on an other method that deals better with non-linear problems and high numbers of features: the random forest algorithm.

5.3.2 Random forest

This time, the theoretical volume difference ΔVol_{th} will not be amongst the explanatory features since the random forest algorithm is supposed to find by itself interesting patterns among features. Therefore, in this whole part, the aim will be to predict ΔVol as a function of $VOLUse1$, H_{eq}^1, ΔH_{eq} as well as the nuclear power plant and the guides' bore and location in the core. Note that the used dataset is the same as the one used in the last two attempts to build a linear model.

First attempt The dataset is divided into a training and a test sets, each one of a size of 80% and 20% of the dataset's size respectively. Then, the random forest is built with default settings, namely a number of grown trees of 500 and a number of variables randomly sampled as candidates at each split of $\frac{p}{3}$ where p is the total number of features. Once the training done, a first thing to observe is the importance of each variables. This information is provided in Figure 5.8 that show the total amount of purity gained by each variable while growing the trees.

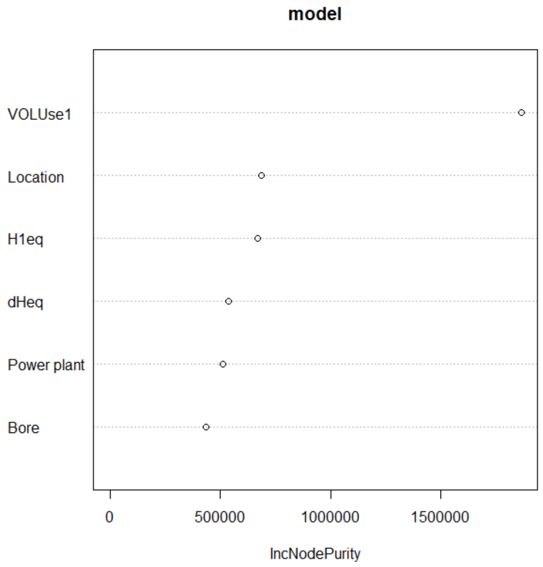


Figure 5.8: Variables importance

This variables importance plot shows that the worn volume at the time of the first control is a fundamental information to predict the difference of worn volume between the two controls, as expected. Nonetheless, it also highlights that H_{eq}^1 and ΔH_{eq} are not as important as $VOLUse1$. Therefore, the choice of directly using the theoretical volume difference ΔVol_{th} to explain the evolution of ΔVol in the linear regression part might not be as relevant as it seemed. Then, no other categorical feature seems to be less relevant than the others.

For this first attempts the obtained R^2 coefficient is of 0.374. Let us try this model with the test set.

The graph on Figure 5.9 shows the predicted values for the whole test set along with their 90% prediction intervals. About 75% of the estimates are positive, while in theory all of them should be greater or equal than 0.

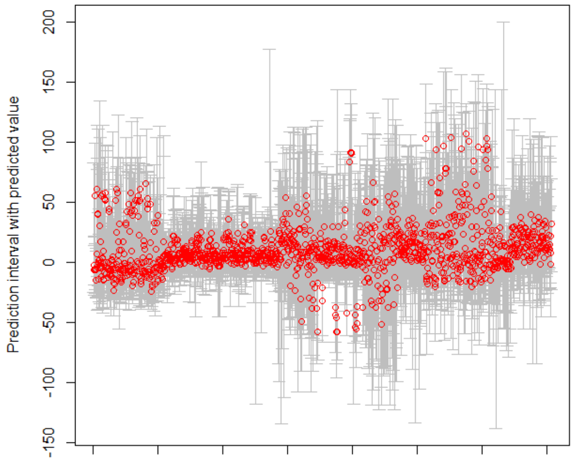


Figure 5.9: Predicted value along with their 90% prediction intervals

Let us now analyze the prediction intervals in detail. To do so, the prediction intervals as well as the true values of ΔVol are centered toward zero. Then, they are ordered from the lowest prediction interval to the highest one and plotted. The result is shown on Figure 5.10.

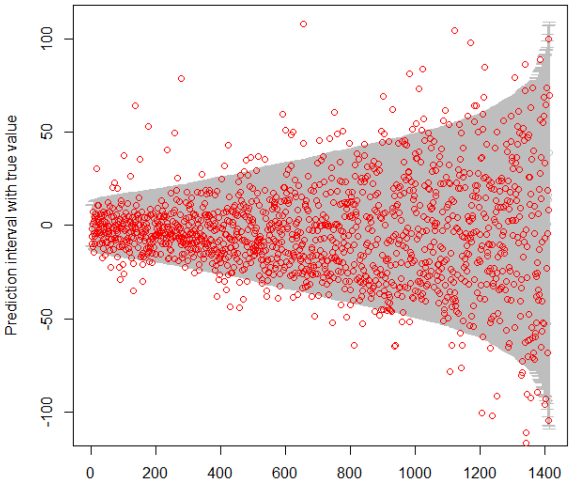


Figure 5.10: Centered true value along with their corresponding 90% prediction intervals

The first observation is that despite the dispersion, more than 90% of the real values are in their corresponding prediction interval. This consolidate the choice of using quantile regression forests to perform these intervals. The second thing to notice is that the size of the intervals, although acceptable on the left of the graph, increases a lot on the right hand side. This leads to difficulties when it comes to estimating precisely ΔVol. Obviously, knowing the rather low obtained R^2 coefficient, one could not expect better results.

Does changing the default settings improve the outcome ? Table 5.5 contains several tests that have been carried out.

Nb. trees / **Nb. candidates**	**100**	**200**	**500**	**1000**
2	0.351	0.369	0.374	0.374
3	0.349	0.362	0.371	0.372
4	0.351	0.354	0.359	0.359

Table 5.5: R^2 values for different settings : the number of candidates is the number of variables that are randomly chosen and tested at each split

Hence, there seems to be no reason to modify the default settings. Note that no cross-validation is needed in the case of random forests since the out-of-bag (OOB) error works in a similar way.

5.4 Practical test

Maintenance has been effectuated on 3 power stations recently and some guides have been changed, some of them as preventive measures using the old model and others due to deterioration. The aim here is to compare the guides that should be changed according to the random forest predictions and the one that have been changed due to deterioration. Remind that a cluster guide is changed if 5 consecutive positions reach the worn volume threshold of 328mm^2.

To estimate the needs of changing a control cluster, the procedure contains the following steps:

1. for each guide position, compute the prediction interval $[q_5, q_{95}]$ for ΔVol
2. add the volume at the time of the first control: $[Q_5, Q_{95}] = VOLUse1 + [q_5, q_{95}]$
3. for each guide position, create two boolean features: $test_5 = (Q_5 > 328)$ and $test_{95} = (Q_{95} > 328)$
4. finally, classify the guides into three categories:
   (a) 2, if $test_{95}$ is true for at least five consecutive positions
   (b) 1, if $test_5$ is true for at least five consecutive positions
   (c) 0 otherwise

The same dataset as before is used. Therefore, since the test set here contains every guides situated in the corresponding 3 power stations, the random forest cannot train using the power station feature. Thus, the results might not be as good as in the previous part. A part of the result is shown in Figure 5.6.

ID	Q_5	Q_{50}	Q_{95}	$test_5$	$test_{95}$
AAA1 A01 E1 GD01	236.03	275.54	317.30	FALSE	FALSE
AAA1 A01 E1 GD02	274.88	308.80	360.36	TRUE	FALSE
AAA1 A01 E1 GD03	346.91	377.02	438.02	TRUE	TRUE
...	...	...	...	...	...

Table 5.6: Prediction intervals and criterion test

Once the prediction intervals built and the criterion tests done, the cluster guides are classified into the three categories previously introduced.

The issue here is that up to this day the guides are changed if there is a risk that the 328mm^2 threshold is reached before the next maintenance. Therefore, the obtained worn volume are inferior to the ones used at the time of the last maintenance. Nonetheless, 100% of the guides classified as 1 or 2 have been changed and only the upper 5% of the guides classified as 0 has been changed, which reinforces, at the very least, the choice of this model.

5.5 Discussion

The aim of this mission was to develop a tool capable of realistically estimating the number of cluster guides that may need to be replaced before a maintenance operation. A simple linear model could have sufficed in the case of noiseless data. However, in an industrial context where each measurement is carried out by a technician with a tool that may differ from one control to another and where the accuracy is not always optimal, noise is automatically generated. Therefore, a noise treatment or most advanced regression methods must be used.

In the current state of affairs, the tool developed during this study cannot be industrialized, due to a lack of precision when estimating the volume worn out. Nevertheless, a campaign of measures in which noise reduction from one measure to another would be the main issue would undoubtedly make it possible to use this tool in a sustainable manner.

In view of the simplicity of the current model for estimating a physical phenomenon as complex as wear and tear, a model using machine learning such as random forests would probably improve predictions and thus reduce maintenance costs. Finally, this study shows the complexity of data analysis in an industrial and complex environment such as nuclear power, where a number of physical, human and material phenomena impact each other and create noise in the data.

Chapter 6

Conclusion

During my internship, four missions were carried out.

Firstly, it could be shown that the maintenance period for the pneumatic valve diaphragms used so far is good based on all the data available since the commissioning of nuclear power plants. However, it is theoretically possible to adapt this maintenance period to each type of valve in order to optimise costs and downtime.

Next, a program was implemented to automate the analysis of the falling time curves of the control clusters. This algorithm offers a combination of outlier detection and classification algorithms as well as a graphical method to make it very accessible. It is ready to use and all that remains to be done is to feed the library of known curves for all power levels in order to make it usable by the agents.

A preliminary study concerning the implementation of a possible incident prediction algorithm on pneumatic valves was also conducted. It identified two indicators that seem to be relevant for the detection of this type of event. EDF, convinced by the proposed approach, is going to implement this algorithm in collaboration with R&D.

Finally, a study aimed at proposing a wear prediction model for the cluster guides was carried out. Despite the noisy data, the use of a quantile regression forest has made it possible to calculate prediction intervals that will then be used to predict the guides to be changed during future maintenance. Furthermore, if a way is found to reduce the initial noise in the data, the machine learning approach seems to be well suited to this type of problem.

This master thesis, which I did as a Data Scientist internship at EDF, allowed me to discover different possible uses of data science in the context of nuclear power plant maintenance and industry in general as well as the difficulties that can get in my way when it comes to analysing data from industry or production.

I had the opportunity to present all this work to specialists in nuclear power plant maintenance and members of the Data Analytics team. Satisfied with the results, more studies of this type should be carried out for the maintenance of nuclear power plants in the coming months and years.

Bibliography

[1] Clifford Anderson-Bergman. icenReg: Regression models for interval censored data in R. *Journal of Statistical Software*, 81(12):1–23, 2017.

[2] Ana Arribas-Gil and Juan Romo. Shape outlier detection and visualization for functional data: the outliergram, 2013.

[3] Ertugrul Colak, Hulya Ozen, Busra Emir, and Setenay Oner. Pairwise multiple comparison adjustment procedure for survival functions with right-censored data. *Computational and Mathematical Methods in Medicine*, 2017:1–8, 10 2017.

[4] Padraig Cunningham and Sarah Delany. k-nearest neighbour classifiers. *Mult Classif Syst*, 04 2007.

[5] Marie Laure Delignette-Muller and Christophe Dutang. fitdistrplus: An R package for fitting distributions. *Journal of Statistical Software*, 64(4):1–34, 2015.

[6] K. Hechenbichler and Klaus Schliep. Weighted k-nearest-neighbor techniques and ordinal classification. *discussion paper*, 399, 01 2004.

[7] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1:14 – 23, 01 2011.

[8] Eve Mathieu-Dupas. Algorithme des k plus proches voisins pondérés et application en diagnostic. In *42èmes Journées de Statistique*, Marseille, France, France, 2010.

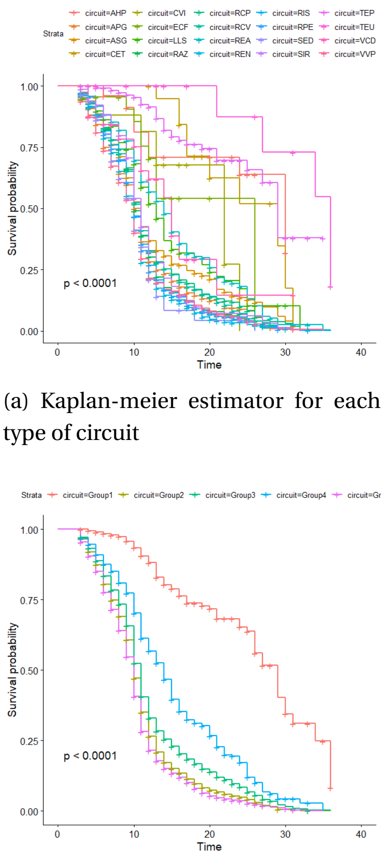
[9] Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999, 2006.

[10] L. Sangalli, P. Secchi, S. Vantini, V. Vitelli, and June. Mox – report no . 13 / 2008 k-means alignment for curve clustering. 2008.

[11] DAVID SCHOENFELD. The asymptotic properties of nonparametric tests for comparing survival distributions. *Biometrika*, 68(1):316–319, 04 1981.

Appendix A

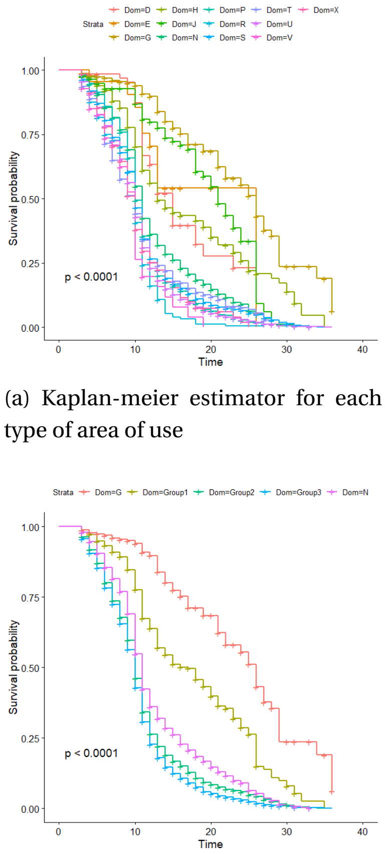
Appendix: Kaplan-Meier



(a) Kaplan-meier estimator for each type of circuit

(b) Kaplan-meier estimator for each group of circuit

Figure A.1: Kaplan-Meier estimator before and after clustering by circuit using pairwise Logrank test



(a) Kaplan-meier estimator for each type of area of use

(b) Kaplan-meier estimator for each group of area of use

Figure A.2: Kaplan-Meier estimator before and after clustering by area of use using pairwise Logrank test