Detection of Anomalous Patterns in Cork Stoppers Sorting Machines

Edgar Gomes Moura

Master's Dissertation

Supervisor at FEUP: Prof. Jorge Pinho de Sousa



Mestrado Integrado em Engenharia e Gestão Industrial

2021-03-25

Abstract

In an era where consumer power is consistently increasing, producers across all industries have been making efforts to meet the quality requirements of the customers in order to remain competitive. This is especially true for the wine industry, where the choice of packaging, namely of the closure for the drink's bottles plays a crucial role on the client's perspective of the quality of the wine. This has pushed for stopper producing companies to develop innovative technologies to enhance the product quality, meeting the highest of the quality demands.

Cork has been proven to be best material to produce closures for wine bottles in what regards the conservation of the drink's characteristics and customer quality perception. This dissertation focuses on the continuous improvement of a recent technology developed by the biggest player on the market of cork stoppers. The technology empowers the company with the ability of commercializing stoppers of the highest of quality and free of the chemical contaminant 2,4,6-trichloroanisole found in cork which diminishes the quality of wines when in contact with them. With this technology, the firm is able to classify each of their closures individually based on the concentration of the contaminant. Only the stoppers deemed as fit for sale are then commercialized. This way, an exploratory study of the data produced by the use of that technology was conducted with the goal of obtaining relevant information for the company, namely concerning anomalous patterns present in the data.

In a first phase, an assessment of the quality of data being currently extracted was performed, revealing serious issues of data pollution. These problems create obstacles to the execution of a rigorous data analysis, restraining the achievement of trustworthy results. As such, the main issues of the database were uncovered and described, allowing for the suggestion of some improvements to the procedure of data introduction in the software that the company currently presents.

In a second approach, using a clean dataset obtained from the preprocessing of the raw data, a study of the influence of dataset's several attributes in the final classification results of the stoppers was conducted. As the main result, it was revealed that the pool of equipments that perform the classification of the stoppers is a major factor of influence on the stopper's acceptance as fit for sale or not. This means that the classification of on stopper as contaminated or not greatly depends on which of the equipments it is analysed. This allowed the construction of a ranking of the population of equipments, based on their propensity to accept or not a random stopper. With this ranking, the firm can pay special attention to those equipments that deviate the most from the general behaviour of the group, taking the necessary actions to ensure they run appropriately.

Lastly, using a representative sample of the equipments of the company, the most common anomalous patterns arising in the data were identified. Among other things, the developed approach demonstrated potential in supporting the currently implemented measures the company has to detect eventual classification failures of the equipments. The anomalous patterns were associated with already known issues of the technology, contributing to the further understanding of the most correct use of the equipments and quality control procedures.

Resumo

Deteção de Padrões Anómalos em Máquinas de Classificação de Rolhas de Cortiça

Numa era em que o poder do consumidor se apresenta em contínuo crescimento, empresas de todos os setores industriais têm vindo a fazer esforços substanciais para responder às exigências de qualidade dos consumidores de modo a manterem-se competitivas. Este facto é especialmente verdadeiro para a indústria vitivinícola, onde a escolha do embalamento dos vinhos, nomeadamente da rolha para selar a garrafa, tem um papel crucial na perceção da qualidade do produto por parte do cliente. Em resultado disso, as empresas produtoras de rolhas para garrafas de vinho têm vindo a desenvolver tecnologias inovadoras de modo a garantir um alto nível de qualidade do seu produto, as rolhas.

A cortiça tem demonstrado ser o melhor material para a produção de rolhas de garrafas de vinho, no que concerne questões da conservação das características da bebida e da perceção da qualidade pelo consumidor. Esta dissertação é focada no contínuo melhoramento de uma tecnologia recente desenvolvida pelo maior competidor no mercado das rolhas de cortiça. Esta tecnologia fornece à empresa a possibilidade da comercialização de rolhas de cortiça da maior qualidade e livres do contaminante químico 2,4,6-tricloroanisol, que pode ser encontrado na cortiça e diminui a qualidade do vinho quando em contacto com a bebida. Usando esta tecnologia, a empresa consegue classificar cada uma das suas rolhas individualmente baseando-se na concentração de contaminante detetada. Assim, apenas rolhas consideradas apropriadas para venda são comercializadas. Posto isto, foi conduzido um estudo exploratório dos dados produzidos pelo uso da tecnologia com o objetivo de obtenção de informação relevante para a empresa, nomeadamente à cerca de padrões anómalos presentes nos dados.

Numa primeira fase, foi efetuada uma avaliação da qualidade dos dados atualmente extraídos, o que revelou sérios problemas de poluição dos dados. Tais problemas criam obstáculos à execução de uma análise dos dados rigorosa, impedindo a obtenção de resultados fiáveis. Assim, os principais problemas da base de dados foram descobertos e descritos, permitindo a sugestão de alguns melhoramentos no procedimento de introdução dos dados no software que a empresa utiliza atualmente.

Numa segunda abordagem, utilizando dados limpos obtidos através do pré-processamento dos dados originais, foi conduzido um estudo da influência de diversos atributos dos dados sobre a classificação das rolhas analisadas. Como resultado principal, ficou estabelecido que o conjunto de equipamentos que efetua a análise das rolhas é um fator dos principais fatores de influência da classificação de cada rolha. Isto significa que as chances de uma rolha ser classificada como aceitável ou não variam bastante conforme o equipamento em que ela é analisada. Isto permitiu a construção de um ranking dos equipamentos, baseado na sua propensão para aceitar ou rejeitar uma rolha aleatória. Com isto, a empresa pode prestar especial atenção aos equipamentos que mostram um comportamento mais díspar da generalidade do grupo, tomando as ações necessárias para assegurar que esses equipamentos funcionam apropriadamente.

Por último, com base numa amostra representativa dos equipamentos da empresa, os padrões anómalos mais comuns que surgem nos dados foram identificados. Entre outras coisas, a abordagem desenvolvida demonstrou grande potencial para suporte das medidas atualmente implementadas pela empresa para detetar potenciais erros de classificação dos equipamentos. Esses padrões foram associados a problemas já conhecidos da tecnologia, contribuindo para um aprofundamento do conhecimento geral do equipamento, do seu correto funcionamento e ainda dos procedimentos de controlo de qualidade a efetuar.

Acknowledgements

My first word of gratitude is directed at Amorim Cork for offering me the invaluable chance to take part on their mission. Only at a great company such as Amorim could I have found a professional, yet familiar, work environment that greatly helped me gain priceless insights on how to conduct professional work. I direct a particular word of acknowledgment to Dr. Miguel Cabral and to Carlos Costa, for the incredible support provided at NDtech and the acceptance of me as any other part of the team. Also, to the Human Resources team I thank, especially to Sara and Joana, for always caring and going the extra mile in ensuring the best conditions, work related or not. I have learned more about the cork stopper business than I imagined possible and I have gained the utmost respect for such a noble product. One of the company's main values is ambition and everyone in the organization, without exception, mirrors that sentiment. I am more ambitious now as well because of this incredibly rewarding opportunity.

To the INESCTEC team, I am thankful for the help provided in this project. Especially to Ricardo Sousa, without whom I could never have done half of what I produced. I am grateful for every single one of the dozens of online meeting hours, for the immense technical support provided, and for being overly helpful in every aspect of this task.

I thank to my supervisor Prof. Jorge Pinho de Sousa for all the knowledge shared with me, for the priceless advices and the continuous availability to contribute for the success of this adventure.

I direct a word of acknowledgment to FEUP, U.D. Sousense and Agrupamento 1328 as well. These are the three institutions that have contributed the most to my growing and development in every aspect. I feel immense proud to be able to say I am part of the history of these institutions as they are part of mine. The values here shared are my values as well.

Additionally, I am very thankful to my friends. First, to all those who have been standing by my side for more years than I can remember, for all the amazing moments shared with you that helped be where and who I am today. Secondly, to the more recently acquired friendships, especially in these last academic years, for all the experiences and common fights for better learning together. Thirdly, to my fellow interns at Amorim, who have truly become friends, for sharing and easing the transition from university to company environment.

A very special word of gratitude to Margarida as well, for being my rock. For every moment of relaxation, all the love and caring, and every moment of distraction from the stress. You are a magnificent treasure.

Lastly, and most importantly I thank my family for putting up with my temper, for believing in me in every circumstance, and for the continuous and unconditional love. You are the best I could ever ask. My thankfulness to you cannot be expressed in mere words.

"In the midst of chaos, there is also opportunity"

Sun Tzu, The Art of War

Contents

1 Introduction **1**
- 1.1 The cork industry . . . 2
  - 1.1.1 Cork stoppers . . . 2
  - 1.1.2 TCA in cork stoppers . . . 3
- 1.2 Amorim Cork . . . 3
  - 1.2.1 NDtech . . . 4
- 1.3 Motivation and aim of the project . . . 5
- 1.4 Document structure . . . 5

2 Literature Review **7**
- 2.1 Machine Learning . . . 8
- 2.2 Time Series . . . 9
  - 2.2.1 Univariate Time Series . . . 9
  - 2.2.2 Multivariate Time Series . . . 10
- 2.3 Data Preprocessing . . . 11
  - 2.3.1 Data summarization . . . 11
  - 2.3.2 Data integration . . . 11
  - 2.3.3 Data cleaning . . . 12
  - 2.3.4 Data transformation . . . 13
  - 2.3.5 Data reduction . . . 13
- 2.4 Feature Engineering . . . 14
  - 2.4.1 Statistical Features . . . 15
  - 2.4.2 Principal components analysis (PCA) . . . 15
- 2.5 Anomalies . . . 15
  - 2.5.1 Types of anomalies . . . 16
  - 2.5.2 Global and local anomalies . . . 17
- 2.6 Anomaly detection . . . 18
  - 2.6.1 Types of anomaly detection techniques . . . 18
  - 2.6.2 Output of anomaly detection techniques . . . 19
- 2.7 Anomaly detection techniques . . . 20
  - 2.7.1 Unsupervised techniques . . . 21
  - 2.7.2 Supervised techniques . . . 22
- 2.8 Sliding Windows . . . 26
- 2.9 Comparison of techniques . . . 27

3 Problem description **29**
- 3.1 The process of detecting TCA . . . 29
  - 3.1.1 Gas chromatography . . . 29
  - 3.1.2 The GC-ECD set-up at NDtech . . . 31

3.2 Quality control . . . 33
3.2.1 Control indicators . . . 33
3.2.2 Pre-analysis control . . . 35
3.2.3 Control during analysis . . . 35
3.2.4 Post-categorization control . . . 36
3.3 Available dataset . . . 37
3.4 Known flaws of the system . . . 38
3.5 Currently implemented measures . . . 39
3.6 Project proposal . . . 40

4 Methodology **43**
4.1 Utilized software . . . 43
4.2 Dataset description . . . 44
4.3 Preprocessing . . . 45
4.3.1 Data Reduction . . . 46
4.3.2 Data summarization . . . 46
4.3.3 Data Cleaning . . . 46
4.3.4 Data transformation . . . 46
4.4 Exploration of the clean variables . . . 47
4.4.1 Relationships between variables . . . 47
4.4.2 Variable influence over the classification of stoppers . . . 48
4.4.3 Selection of the sample . . . 49
4.5 Detection of anomalous patterns . . . 50
4.5.1 Clustering approach . . . 50
4.5.2 Secondary approaches . . . 52

5 Results discussion **55**
5.1 Data preprocessing . . . 55
5.1.1 Data summarization . . . 55
5.1.2 Data cleaning . . . 61
5.1.3 Data tranformation . . . 65
5.2 Exploration of the clean variables . . . 67
5.2.1 Relationships between variables . . . 67
5.2.2 Variable influence over the classification of stoppers . . . 68
5.2.3 Selection of the sample . . . 71
5.3 Detection of anomalous patterns . . . 72
5.3.1 Clustering approach . . . 72
5.3.2 Secondary approaches . . . 74

6 Conclusion and future work **77**
6.1 Main conclusions . . . 77
6.2 Future work . . . 79

A Bar charts and histogramms of raw data variables **85**

B Data cleaning support **101**

C Tables, bar charts and histogramms of the clean data variables **103**

D Relationships between variables **119**

E Anomalous patterns obtained with the clustering approach 139

F Anomalous patterns found with secondary approach based on the 'PeakValue' variable's signal variation 147

G Anomalous patterns found with secondary approach based on the 'Error' variable 151

H Selection of the main sample 153

List of Figures

1.1 Structure of the firm 4

2.1 Steps of the KDD process (adapted from Guerra-Hernández et al. (2008)) 7
2.2 Univariate and Multivariate time series comparison (source: Blázquez-García et al. (2020) 10
2.3 Types of Anomalies 16

3.1 General gas chromatography configuration (adapted from Delloyd's Lab-Tech Chemistry (nd)) 30
3.2 Main steps of the TCA detection cycle in NDtech's modules 32

A.1 Bar chart of raw 'Punit' variable 86
A.2 Bar chart of raw 'Chamber' variable 87
A.3 Bar chart of raw 'PunitLine' variable 88
A.4 Bar chart of raw 'Line' variable 89
A.5 Bar chart of raw 'Shift' variable 90
A.6 Bar chart of raw 'Mode' variable 91
A.7 Bar chart of raw 'State' variable 92
A.8 Bar chart of raw 'Lot' variable 93
A.9 Bar chart of raw 'Class' variable 94
A.10 Bar chart of raw 'Calibre' variable 95
A.11 Histogram of raw 'Peakvalue' variable - all values 96
A.12 Histogram of raw 'Peakvalue' variable - values up to 1000 mV.s 97
A.13 Histogram of raw 'PeakTime' variable 98
A.14 Bar chart of raw 'DeltaTime' variable (bar chart instead of histogram to facilitate visualization) 99
A.15 Bar chart of raw 'Quality' variable 100

C.1 Bar chart of clean 'Module' variable (previously called 'PUnit') 104
C.2 Bar chart of clean 'Chamber' variable 105
C.3 Bar chart of clean 'Line' variable 106
C.4 Bar chart of clean 'Shift' variable 107
C.5 Bar chart of clean 'Lot' variable 108
C.6 Bar chart of clean 'Class' variable 109
C.7 Bar chart of clean 'Calibre' variable 110
C.8 Histogram of clean 'Peakvalue' variable - all values 111
C.9 Histogram of clean 'Peakvalue' variable - values up to 1000 mV.s 112
C.10 Histogram of clean 'PeakTime' variable 113

C.11 Bar chart of clean 'AnalysisDuration' variable (previously called 'DeltaTime'; bar chart instead of histogram to facilitate visualization) . . . 114
C.12 Bar chart of clean 'Quality' variable . . . 115
C.13 Bar chart of the created 'Origin' variable . . . 116
C.14 Bar chart of created 'VSR' variable . . . 117
C.15 Bar chart of created 'Error' variable . . . 118

D.1 Percentage of classified G1 stoppers per Module . . . 120
D.2 Percentage of classified G2 stoppers per Module . . . 121
D.3 Percentage of classified G3 stoppers per Module . . . 122
D.4 Stacked percentages of stoppers classified for each group per Module . . . 123
D.5 Average PeakValue per Module . . . 124
D.6 Percentage of observations with error per Module . . . 125
D.7 Percentage of classified G1 stoppers per Line . . . 126
D.8 Percentage of classified G2 stoppers per Line . . . 127
D.9 Percentage of classified G3 stoppers per Line . . . 128
D.10 Stacked percentages of stoppers classified for each group per Line . . . 129
D.11 Average PeakValue per Chamber . . . 130
D.12 Percentage of classified G1 stoppers per Lot . . . 131
D.13 Percentage of classified G2 stoppers per Lot . . . 132
D.14 Percentage of classified G3 stoppers per Lot . . . 133
D.15 Stacked percentages of stoppers classified for each group per Lot . . . 134
D.16 Average PeakValue per Lot . . . 135
D.17 Percentage of classified G1 stoppers per Origin . . . 136
D.18 Percentage of classified G1 stoppers per type of lot (VSR or non-VSR) . . . 137

E.1 8G2 pattern . . . 140
E.2 4G2 pattern . . . 141
E.3 Integration error pattern . . . 142
E.4 Isolated Peak pattern . . . 143
E.5 Cyclic CO pattern . . . 144
E.6 Chamber CO associated to Cyclic CO pattern . . . 145

F.1 Frequency of each type of variation in two consecutive cycles . . . 148
F.2 Frequency of each type of variation in four consecutive cycles . . . 149

G.1 Most common sequence of errors in one cycle . . . 152

H.1 Acceptance of a sample of lots by different modules . . . 155
H.2 Graphical representation of the module ranking . . . 156

List of Tables

4.1 Brief description of the relevant features 44

5.1 Summary of variable 'AnalTime' 56
5.2 Summary of variables depicting conditions of analysis 58
5.3 Summary of variables depicting characteristics of the stoppers 59
5.4 Summary of the numerical variables that shows the resultant classification of the chromatographic analysis ('PeakValue' is measured in mV.s; 'PeakTime' and 'DeltaTime' are in seconds) 60
5.5 Summary of the categorical variable that shows the results of the chromatographic analysis 61
5.6 Actions taken to clean variable 'Line' 63
5.7 Variations of the true calibres in the data 65
5.8 Correspondence between 'Error' variable classes and their description 66
5.9 Significant Mutual Information of categorical variables 67
5.10 Kruskall-Wallis test results where null hypothesis was rejected ($p_{value} > \alpha$) . . . 68

B.1 Correspondence of units to lines according to variables 'PunitLine' and 'Line' . . 102

C.1 Summary of variable clean 'AnalysisDate' (previously called 'AnalTime') 103
C.2 Summary of the clean categorical variables (?module' was previously called 'PUnit') 103
C.3 Summary of the clean numerical variables ('PeakValue' is measured in mV.s; 'PeakTime' and 'AnalysisDuration' are in seconds; 'AnalysisDuration was previously called 'DeltaTime') 103

E.1 Results of application of the clustering technique in each module and each frame 146

H.1 Ranking of the Modules 154

Chapter 1

Introduction

Product packaging today is one of the most relevant features of a product. It is known to influence customer's attraction to a product, pushing companies to innovate the way they package goods (Alhamdi, 2020). For food producers, this is a key aspect of the business. For them, the main function of the package besides its advertising role is to protect the food from getting damaged or spoiled (Silayoi and Speece, 2007), thus preventing any health safety concerns or any unpleasant tasting experience for the final consumer.

The specific case of the wine industry illustrates how important the packaging issue is. Typically, wine is packaged in a glass bottle which is then labelled and capped, and although bottles and labels come in all different shapes and colours, the choice of closure might very well be the most difficult decision for wine producers (Schmitt, 2017).

Wine bottle stoppers can be made from different materials such as cork, plastic, or metal (namely aluminium). Among all, natural cork stoppers are one of the most used by wine producers to seal their bottles. Undoubtedly, natural cork stoppers are used for premium wines, with 89% of the wines in the Wine Spectator Top 100 Wines list being sealed with cork closures (Amorim Cork, 2020a). This preference has paved the way for the growth and development of the cork industry, especially for the cork stoppers market. However, this has not been without its obstacles. As described further ahead, the cork industry has seen serious threats in past years and companies struggled to keep the market growing. Organizations had to adapt and evolve to be able to meet the quality requirements that winemakers and consumers demand. Nowadays, these corporations are highly innovative enterprises using high-end technology to attain their goals.

This dissertation is the result of a project developed at Amorim Cork S.A., a company in the cork industry, with the partnership of the technological research centre INESCTEC. The project focuses on using machine learning techniques to better control one of the company's most important steps in the selection and categorization of their stoppers: categorization based on the detected 2,4,6-trichloroanisole (TCA) concentration.

1.1 The cork industry

Cork is simply the outer bark of the cork oak tree, which is the genesis of all the industry. Although it can grow in different climates, this species flourishes better under the presence of certain environmental conditions typical of the region surrounding the Mediterranean (Cork Quality Council, 2020a). For this reason, countries of this region have the great majority of the world's cork oak forests and consequently are also the biggest producers and exporters of cork (APCOR, 2019a). The Portuguese case can be taken as an example of this. With 34% of the world's cork oak forests (APCOR, 2019a) and more than 60% of the global cork exports (APCOR, 2019b), the country may well be classified as the most invested nation in this industry. Its mediterranean location allows for the existence of vast cork oak forests, naturally leading to the presence of some of the biggest international companies of the sector.

Despite most of the production of cork being concentrated in a limited area of the planet, the industry has become massive. As a sector of the global economy, it has been consistently growing for the past decade and in 2018 the total value of cork exportations in the world amounted to a record 1.808 million euros (International Trade Statistics, 2020). Such an immensity derives from the fact that cork is one of the most versatile materials available in nature. Low density, elasticity and high resistance to abrasion are some of its physical attributes that make it perfect for many different applications. Once removed from the tree, this natural raw material can be converted into the desired products using the appropriate industrial processes. Today, besides its historical use as closure for wine bottles, cork is already used in civil construction, in the textile industry and even in the automobile and aerospace markets (Corticeira Amorim, 2020b).

1.1.1 Cork stoppers

Despite all of its possible applications, most of the produced cork ends up serving the wine industry under the form of a stopper for wine bottles. These are usually defined in two categories: natural and technical/agglomerated. Natural cork stoppers are obtained using a machine to punch the cylindrical stoppers out of cork oak bark strips (Corticeira Amorim, 2020c). They are considered the premium end of cork stoppers because being made out of a single piece of cork allows them to maintain all the best characteristics of the material. The second class of cork stoppers are the agglomerated stoppers, named this way for being produced using shredded pieces of cork that are agglomerated using a mixture of glue and oil suited for products that contact with food.

Wine producers around the globe have been using cork closures to seal their bottles for decades and still prefer them over capsules of other materials. For natural cork stoppers the preference is even clearer and there are plenty of reasons for this:

- Cork's physical characteristics make it a compressible and impervious material, which is regarded by winemakers as perfectly suited for wine preservation (Cork Quality Council, 2020b).

- Consumers tend to perceive wines capped with natural cork stoppers as having higher quality (Reynolds et al., 2018), and it is only natural that producers try to meet the preferences of the end consumer.

- Natural cork stoppers production contributes to the preservation of the environment. According to a study conducted by EY for Corticeira Amorim, producing natural cork stoppers in the company has a negative carbon footprint. For each of these sealants, 309 grams of CO_2 *eq* are absorbed from the atmosphere (EY, 2019). In other words, each of these sealants contributes to the reduction of carbon dioxide in the atmosphere, thus fighting climate change.

All these aspects drive wine producers to privilege the use cork stoppers. On the other hand, though slowly, both wine production and consumption have been increasing worldwide since the beginning of the century, reaching 292 and 246 million hl, respectively in 2018. Together, these two factors have created a great demand for cork stoppers. According to APCOR (2019b), 70.5% of the total exported cork from Portugal in 2018 was under the form of stoppers (41.6% of which were natural cork stoppers). This is valued at an astonishing 752.7 million euros.

It may be argued that this is a very simple product, but it has been the target of many studies making it a very versatile and innovative product, being now used not only for wine and champagne bottles but for spiritual beverages such as gin, whiskey or vodka as well.

1.1.2 TCA in cork stoppers

As previously mentioned, the history of the industry shows that companies had to overcome some obstacles to get the business growing. Particularly, during the 90s a mistrust towards cork sealers took place. At the root of this problem was 2,4,6-trichloroanisole, a chemical compound that can be found in cork and consequently in cork stoppers. When it comes into contact with wine, this compound can taint the drink, giving it a mouldy odour and flavour, commonly known as "cork taint".

Naturally, neither customers would want to buy a spoiled product nor did the companies want to sell such a product. This led to a shift in preference for capsules made from other materials on the part of wine makers. Nonetheless, cork stopper companies invested in research and innovation and nowadays processes have been created to control, purify, or discard TCA contaminated stoppers. This has rebounded cork stoppers back to the spotlight of wine bottle capsules, returning to being the undisputed number one choice.

1.2 Amorim Cork

Corticeira Amorim is the largest company in the cork industry. With a global presence, the company exports cork and cork products to clients all around the world. Founded in 1870, the Portuguese company started as a cork stoppers production company in Vila Nova de Gaia. Since then,

it has been growing and expanding its portfolio of businesses and products (Corticeira Amorim, 2020a). The current structure of the company can be observed in Figure 1.1, where the various business units are observable.

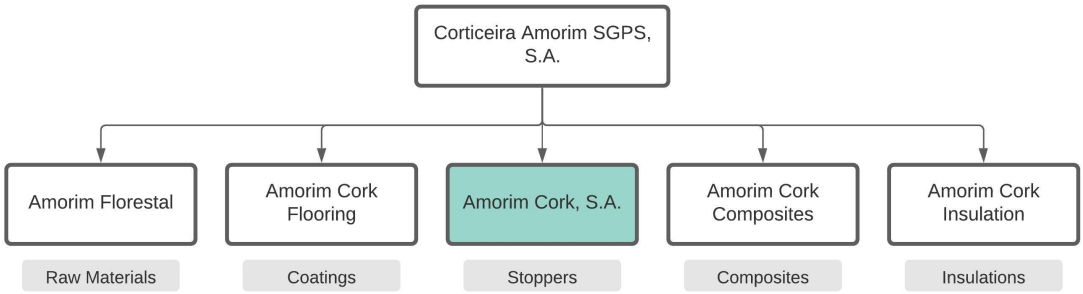


Figure 1.1: Structure of the firm

Amorim Cork S.A. is the business unit responsible for the entire operation regarding cork stoppers. It is divided in several industrial units, each responsible for a different phase of production of their cork stoppers. At one of those industrial units, Portocork, a special technology was developed to address the issue of TCA – NDtech.

1.2.1 NDtech

Portocork is not a traditional manufacturing company. They operate in a buy-and-resell basis, buying cork stoppers from other industrial units of the group or from external suppliers, and reselling them later. NDtech is the department of Portocork where the homonym technology is used. It is the company's strategic response against the TCA issue, and with this advanced technology Portocork is able to measure the TCA concentration levels in each stopper they buy. This way, they can select and separate stoppers that have humanly undetectable quantities of TCA from contaminated stoppers. By doing this separation the company can be sure of selling only stoppers that will not contaminate the wine.

The technology works on the basis of chromatography, which is a chemical analysis process that allows for the separation of the different chemical components on a given sample. Gas chromatography is a specific type of chromatography, which, as the name indicates, separates the chemical components carried by a gas flow. It is the type of chromatography used at NDtech.

Simply put, each stopper is isolated in a camera where a gaseous nitrogen flow drags the chemical components out of the cork's surface. This mixture of nitrogen and chemical compounds is then separated in a hollow column (which is the principle of chromatography). By separating the TCA from other compounds, the system is then able to accurately read the concentration of contaminant in the stopper. Finally, based on this measurement the stoppers are classified and sorted into the appropriate group. A more detailed description of the complete process will be given in Chapter 3.

1.3 Motivation and aim of the project

As already mentioned, the key issue being addressed in this document is the presence of TCA in cork stoppers. The main goal of NDtech's technology is to empower the company with the possibility of selling cork stoppers to customers with the assurance that none of the corks has detectable amounts of TCA. However, as sophisticated as this technology is, the fact that it is very recent and innovative means that there are many potential issues with the system that must be addressed. Therefore, strategies must be developed in order to better control the process, ensuring that the system correctly measures the contaminant's quantity in each stopper.

This is of extreme importance, once an incorrect reading can have one of two consequences. Either the TCA amount is underrated and so the company will sell contaminated stoppers, ultimately leading to mistrust on the part of the clients. Or the measurement is overrated, and the company will discard stoppers in good conditions, which in other words corresponds to losing money.

Considering this, one approach to address potential errors in the equipments is to collect and analyse data produced by the system's measurements. Utilizing Machine Learning techniques, namely anomaly detection techniques, computerized algorithms can go through the data and look for anomalous patterns in the data. These patterns may be indicative of potential failures and reading defects of the system, and once the detection of such patterns is done in real time, the company will be able to promptly reconfigure the machines as soon as a problem is identified, preventing misclassification of the cork stoppers.

Therefore, an exploratory study of the intrinsic patterns that arise in data extracted from the use of NDtech's technology is of great interest to the company. Such study is the aim of this work. It will allow for the identification of patterns that are anomalous and thus indicate that there is some deviation in the system's behaviour. This discovery represents an opportunity for the organization to correct and prevent flaws and failures of the machines, ultimately increasing the reliability of the technology.

1.4 Document structure

In this introductory chapter a general presentation of the theme was presented. The company, the project, and the problem in question were highlighted, as well as the general market. Furthermore, the core objective of this work has also been described. Chapter 2 exposes the existing literature on Machine Learning techniques focused on pattern identification. Next, Chapter 3 thoroughly describes the problem in question, mentioning how the technology works and the issues that need to be addressed. In Chapter 4, a description of the methods used to evaluate and solve the problem is presented. The results of the application of these methods are then reported and discussed in Chapter 5. Lastly, Chapter 6 displays an overview of the core conclusions, suggesting as well, additional work to be executed in the future.

Chapter 2

Literature Review

As mentioned in the previous chapter, the main objective of this work is to construct algorithms capable of detecting and cataloguing anomalous patterns in the data extracted from machines at NDtech to allow the company to make decisions regarding the machine's inspections and corrections. In other words, knowledge must be extracted from data to facilitate decision making. This corresponds to the description of the Knowledge Discovery in Databases process (KDD). In short, KDD consists of collecting and selecting data, then preprocessing it and finally applying Machine Learning (ML) techniques to acquire knowledge (in this case, to find anomalous patterns), as shown if Figure 2.1.

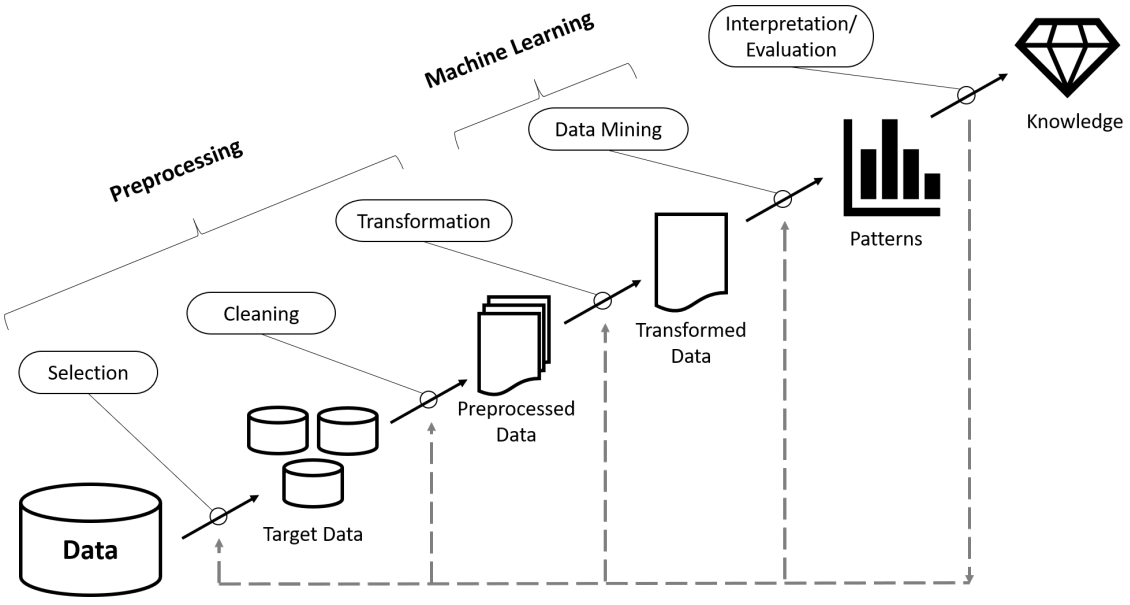


Figure 2.1: Steps of the KDD process (adapted from Guerra-Hernández et al. (2008))

The sequence of events that ultimately lead to the detection of anomalies is quite clear. Firstly, data must be acquired. Sensors capture/measure the characteristics of the system under analysis and send its readings to a database. This is the data collection step. Next, comes the preprocessing stage. At this point data is inspected to verify if it is presented as expected. In other words, the

inspection is done to guarantee that data is clean and in pre-established formats in accordance to the system's behaviour. Once inspected and corrected, the dataset is ready to be processed, hence the name preprocessing. After this, typically there is an optional Feature Engineering (FE) step which serves to extract the relevant information provided by the data in a more useful way. This can be seen as an integrating part of data preprocessing, or as a subsequent step. Finally, with the data ready to be analysed, there comes the actual processing phase where Machine Learning techniques are used to perform the analysis, which in our case is to detect anomalous patterns. This last step may be further divided in training and testing for example, which will be mentioned ahead (Shanthamallu et al., 2018; Agrawal and Agrawal, 2015).

With this in mind, the remainder of this chapter is organized in the following way: first, the concepts of Machine Learning is introduced, with mentions to Data Mining and Data Science. This is presented in first place because these are broad concepts that respect and influence what is described in the rest of the chapter. Next, the specific type of data produced at NDtech, Time Series (TS), is presented. Having described the type of data at play, the various stages of the KDD process are introduced, starting with the preprocessing phase and each of its steps mentioned. After that, the Feature Engineering phase is presented. Next, the definitions of anomalies and anomaly detection are presented in order to detail the available ML techniques used for detecting patterns and anomalies in TS data, including examples from the available literature. Finally, the chapter closes with general considerations of the reviewed approaches, exposing some of their respective merits and weaknesses.

2.1 Machine Learning

Literature has defined Machine Learning as a field of study which enables computers to learn without being explicitly programmed by a human. In other words, ML refers to the concept of machines learning from past experience to improve future performance of the processes they are responsible for. This learning process is accomplished through experience with examples, that is, data samples (Shanthamallu et al., 2018). This way, when exposed to new data, computers re-program themselves on the basis of some predefined strategy (Das et al., 2017).

Data Mining (DM) is another important concept that is often assumed to be the same as ML. Though the two subjects often overlap, there are some differences between both. More specifically, there can be established a sense of hierarchy between ML and DM. Data Mining is about analysing and interpreting data to retrieve useful information. This information may then be used as basis for Machine Learning to perform its automatic learning process, training and modifying the machine (Das et al., 2017).

Consequently, both ML and DM fall under the umbrella of Data Science (Import.io, 2017), which, as the name indicates, is the science that studies data and its correlations and attributes, revealing its features and hidden structures (Hayashi, 1998). In other words, Data Science studies and explores the properties of data with the purpose of extracting knowledge. It involves all the

steps from generating/acquiring data to the form of exhibit to the users, passing through the preprocessing and processing of data. Machine learning feeds of Data Mining techniques to accomplish the learning process, thus enabling the machine to re-program itself based on its understanding of the input data and its features.

2.2 Time Series

It is also essential to understand the type of data analysed under the scope of this dissertation. As detailed ahead in Chapter 3, each record of the dataset corresponds to a measurement of the quantity of TCA on a single cork. This way, every time a cork goes through the system, it creates a new data point with several attributes, including the TCA concentration measurement and the time (date and hour) at which the observation was recorded. Being this a continuous process, it becomes evident that each data point is defined, among other factors, as a function of time, which in turn allows us to classify the data as a Time Series.

Time Series data is the concept of a dataset composed of observations recorded under a certain time sequence or interval (Liu et al., 2020), or, in other words, a set of observations collected sequentially in time (Basu and Meckesheimer, 2007). This way, time series are part of a broader class of data, sequence data. Generally, a sequence is an ordered series of events, where data instances are related to each other. Although sequence data can be of different types, such as binary, discrete, or continuous, data points always follow a sequential order. Continuous sequences (or time series) are one of the most common form of sequences found in real life applications (Chandola et al., 2012, 2009).

An important characteristic of TS is its dimensionality. This describes the number of separate data attributes each observation presents (Cook et al., 2020). Therefore, time series can be composed of a single time-dependent variable or multiple time-dependent variables. The first type of TS is referred to as Univariate Time series (UTS) and the latter as Multivariate Time series (MTS).

2.2.1 Univariate Time Series

In practical terms, time series represent the output from a single source (a sensor) linked to the time of the observation. This stream of data has often the form of a key-value pair, where the timestamp is the key and the value is a scalar, nominal or ordinal reading, x_t (Cook et al., 2020). A more formal definition for UTS is provided by Blázquez-García et al. (2020):

A "univariate time series $X = \{x_t\}_{t \epsilon T}$ is an ordered set of real-valued observations, where each observation is recorded at a specific time $t \epsilon T \subseteq \mathbb{Z}^+$".

Thus, "x_t is the point collected at time t and $S = x_p, x_{p+1}, ..., x_{p+n-1}$ the subsequence of length $n \leq |T|$ starting at position p of the time series X, for p, t ϵ T and $p \leq |T| - n + 1$. It is assumed that each observation x_t is a realized value of a certain random variable X_t".

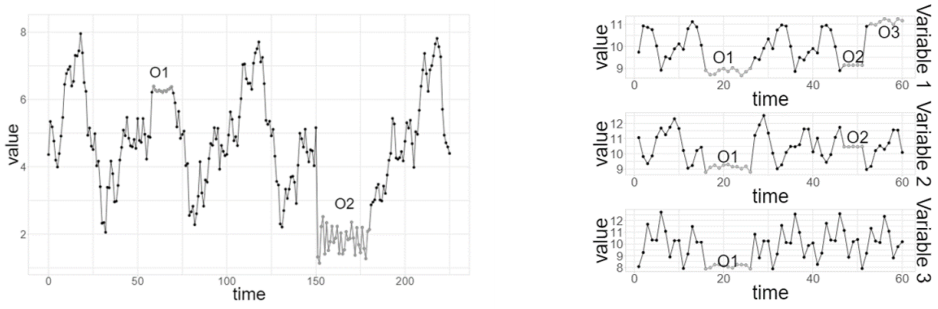
2.2.2 Multivariate Time Series

Contrasting with univariate time series, multivariate data consists of a sequence of observations taken by multiple sensors. This stream of data is usually in the form of a key-vector pair, where a number of observations are taken at the same time, each associated with a different sensor reading the system, $x^t = [x_1^t, ..., x_n^t]$. These may be understood as temporally associated univariate data streams that offer a more complete view of the system (Cook et al., 2020). Another way of looking at MTS is as a set of data streams that are correlated. Again, Blázquez-García et al. (2020) provide a formal definition for MTS:

"A multivariate time series $X = \{x_t\}_{t \epsilon T}$ is defined as an ordered set of k-dimensional vectors, each of which is recorded at a specific time $t \epsilon T \subseteq \mathbb{Z}^+$ and consists of k real-valued observations, $x_t = (x_{1t}, ..., x_{kt})$".

Thus, "x_t is said to be a point and $S = x_p, x_{p+}1, ..., x_{p+n-1}$ a subsequence of length $n \leq |T|$ of the multivariate time series X, for p, $t \epsilon T$ and $p \leq |T| - n + 1$. For each dimension $j \epsilon \{1, ...,k\}$, $X_j = \{x_{jt}\}_{t \epsilon T}$ is a univariate time series and each observation x_{jt} in the vector x_t is a realized value of a random time-dependent variable X_{jt} in $X_t = (X_{1t}, ..., X_{kt})$. In this case, each variable could depend not only on its past values but also on the other time-dependent variables".

Considering this, it is possible to infer that performing Anomaly Detection (AD) has some differences depending on the kind of time series being used. Anomaly detection over univariate time series leans on the comparison of the current observation against information provided by the history of the sole time series. When dealing with multivariate time series, besides the history of the stream, the relationship between each of the variables which combine to form the observation is pivotal for the detection of the anomaly. This may allow the anomaly detection method to perform more accurately by utilizing the additional contextual information provided by multiple monitoring approaches. Furthermore, the dimensionality of data will also influence the choice of AD technique since some techniques perform poorly with high dimensional data. The computational complexity posed by MTS may complicate the task, especially when in an online decision context (Cook et al., 2020). Figure 2.2 illustrates the difference between UTS and MTS.



(a) Univariate time series

(b) Multivariate time series

Figure 2.2: Univariate and Multivariate time series comparison (source: Blázquez-García et al. (2020)

2.3 Data Preprocessing

As the first step in the Knowledge Discovery Process after the data is collected, data preprocessing is very important and presents very clear reasons for having been developed as an independent stage of data analysis. In the real world, raw data is highly vulnerable to imperfections. Data can be incomplete, i.e., with missing values (e.g. Atr_a = ' '), inconsistent (e.g. Atr_a = '1', Atr_a = 'one'), noisy (e.g. negative value in a distance measure), and/or redundant (e.g. duplicate attributes) (Alasadi and Bhaya, 2017). The presence of such issues may seriously undermine the performance of subsequent learning processes and algorithms, resulting in misleading analysis. No quality data, no quality results, and therefore, no quality decisions (Yang, 2013). Because of this, Data Preprocessing is a very important task which aims at transforming raw data into high quality data with a more understandable structure and which properly fits the analysis processes to follow. It is an activity that improves the efficiency and usefulness of data, consequently improving the precision and speed of posterior learning processes, and ultimately enhancing the reliability of subsequent discoveries (Ramírez-Gallego et al., 2017).

With the focus on multiple tasks, as reducing data size, finding relations between attributes, normalizing data, removing outliers and extracting features from data, DP is one of the most time-consuming tasks for an analyst, taking up to 60% of all the efforts related to a certain project (Yang, 2013). This multidisciplinary side of DP allows for a subdivision of tasks inside it. Typically, data preprocessing can be divided into integration, cleaning, transformation and reduction, for example (Alasadi and Bhaya, 2017). Data summarization (based on statistics) can also be included as a first task which helps understanding exactly what must be done in all the others steps. A brief description of each of them follows bellow.

2.3.1 Data summarization

As mentioned, this serves as a first step before the actual tasks of data preprocessing, helping the analyst to understand the data by providing information such as the trends, variations and spread of the data. This can be accomplished by calculating some dispersion measures such as median, mean, max, min, quantiles, variance, etc. Also, some basic graphic displays may be produced such as histograms, boxplots, and scatter plots to help the comprehension of the dataset (Yang, 2013).

2.3.2 Data integration

Data integration deals with combining data from different sources into one consistent database (Alasadi and Bhaya, 2017). Doing this requires attention to some details to avoid redundancies and inconsistencies in the integrated data. Therefore, the multiple sources must be verified to identify some issues. For example, the same record may be present in more than one source, which could lead to duplicated records. A careful integration of data helps reducing redundancies and inconsistencies in data, improving quality and speed of future analyses (Yang, 2013). However, in

the case of this project, a unique database is used, and, therefore, this step does not apply to this work.

2.3.3 Data cleaning

Data cleaning serves the purpose of filling missing values, smoothing noisy data, identifying and removing damaged/spurious values and outliers, and resolving inconsistencies (Alasadi and Bhaya, 2017).

Regarding missing values, many times an object does not have a recorded value for one or more of its attributes, which can be caused by equipment malfunctions or data not being entered due to misunderstandings, among other reasons (cite Yang 2018). This can be handled in different ways:

- Ignoring it – which often is not done because the absence of values may lead to incomplete analysis;
- Filling it manually – which is very tedious and infeasible in many situations, especially in large data sets with several missing values;
- Filling it automatically – which is perhaps the smarter and most popular choice. It can be done in different ways:
  - Using a global constant, for example '*Unknown*', though in numerical attributes, for example, it is not a good option;
  - Using the mean or the median of the attribute;
  - Using the mean or median of the attribute for only all instances belonging to the same class. In other words, use the mean of all observations which share the same value for another attribute;
  - Use the most probable value (based on an inference technique).

As for noisy data, this corresponds to random errors or variances in a variable and is a very common issue. Duplicate records, incomplete and inconsistent data are all examples of noise in data. Common roots for this problem include faulty data collection instruments, data entry problems, data transmission problems, technology limitations and inconsistency in naming convention (Yang, 2013). For this issue there are also a few strategies that can utilized.

The first is Binning, which is used to smooth data on its local neighbourhood - local smoothing. This works by first sorting values and then dividing them into 'buckets', that is, bins (Alasadi and Bhaya, 2017). Equal-width and equal-depth (or frequency) are the two ways to partition data into buckets. In the first way, all buckets have the same amplitude of values, regardless of how many observations each one gets. It is the most straightforward strategy, but skewed data is not well handled. In the latter approach, each bin receives the same number of records, regardless of the amplitude of values covered (Yang, 2013). After partitioning, data inside each bin may

be smoothed in different ways as well. Data values can be replaced by the respective bin mean or median, or by one of its bin's boundaries, depending to which boundary, upper or lower, it is closer to (Alasadi and Bhaya, 2017).Another common strategy to smooth data is doing regression. In other words, fitting data to regression functions.

It is important to distinguish between noise removal and anomaly detection. Though both concepts seem very similar and are in fact related, they have distinct defining characteristics. Noise is defined as a phenomenon in data that is not of interest to the analyst but acts as an obstacle to data analysis (Chandola et al., 2009). It can be seen as an event which is not provoked by the system under study, but by external factors. Hence, noise removal aims at eliminating the unwanted objects before any data analysis is performed. Anomaly detection on its side, is part of analysis of data and studies odd/strange values (outliers) assumed to be generated by the system. In other words, anomalies are information.

2.3.4 Data transformation

Data transformation is a general concept whose defining objective is to transform data to forms that are suitable for analysis(Alasadi and Bhaya, 2017). That is, a new form of data that outstands the information of interest. As detailed by Yang (2013), it may involve, among others, the following example activities:

- Smoothing data – already covered and explained under the Data Cleaning task;
- Aggregation or summarization. This consists of applying statistical measures to summarize data. For example, to turn a monthly attribute into a daily attribute using its mean;
- Attribute construction. This corresponds to creating new attributes based on existing ones, to ultimately replace the old ones (this overlaps with Feature Engineering, which is addressed below).
- Normalization. This translates into scaling data for it to fall within a specified range, for example between 0 and 1. The most popular forms of normalization are:
  - Min-max normalization: $v' = \frac{v - min_A}{max_A - min_A}(new_max_A - new_min_A) + new_min_A$
  - Z-score normalization: $v' = \frac{v - mean_A}{stand_dev_A}$
  - Decimal scaling: $v' = \frac{v}{10^j}$, where j is the smallest integer such that Max($|v'|$)< 1

2.3.5 Data reduction

Data reduction, as the name indicates, aims at reducing data representation in volume, while maintaining the integrity and useful information of the original dataset and ensuring the reduced dataset produces similar analytical results (Alasadi and Bhaya, 2017). It is performed because databases may store terabytes of data over which, complex data analyses may take a very long

time to run for the entire dataset. By employing data reduction strategies, one spares computational resources and time while maintaining the integrity of the analyses. These strategies include dimensionality reduction and numerosity reduction (Yang, 2013).

Dimensionality reduction or Feature Selection is aimed at reducing the number of attributes of a data set. This can be done in a plurality of forms. A simple manual feature selection procedure can be performed, where one can use common sense and previously acquired knowledge of the data to deduce if there are any features that are clearly unnecessary. Another way to reduce the number of attributes is employing a technique to perform this feature selection. This can be done utilizing a heuristic method to determine which features or combination of features are the most important to keep for analysis and which are redundant and should be removed. It can also be done using statistical and transformation-based methods, such as PCA or Wavelet Transformation (Alasadi and Bhaya, 2017), or with algorithms such as Projection Pursuit, Independent Component Analysis (ICA) and Relief (Cook et al., 2020; Blázquez-García et al., 2020). The new variables may provide insights into hidden states within the data, previously impossible to observe with the original data (Cook et al., 2020). Additionally, because the new series corresponds to combinations of the initial input variables, any identified anomalies in the new dataset are multivariate (they affect more than one of the initial variables) (Blázquez-García et al., 2020). This largely overlaps and intertwines with the task of Feature Engineering, which will be addressed ahead.

Numerosity Reduction corresponds to reducing data volume in massive datasets by choosing alternative, smaller forms to represent data. The most popular way to achieve this is by sampling data. This non-parametric method is used to obtain a small sample to represent the entire database, allowing ML algorithms to tun with much lower complexity. The sampling procedure may be performed randomly, or following a certain logic, usually stratified sampling, which guarantees the sample presents approximately the same percentage of instances belonging to each class of the original database.

2.4 Feature Engineering

Feature Engineering or Feature Extraction is used to obtain new features from existing ones in a dataset. In many cases, not all variables that are present in a dataset help explain the behaviour of a system. For this reason, a strategy to determine which are the most important features in a data set is needed. Besides, it is also important to obtain new sets of attributes that better capture the underlying behaviour of the system and provide more useful information to analysts, and the combination of original and extracted features that reduce the complexity of the problem and achieve the most reliable posterior analysis (Men et al., 2018). This description overlaps with some concepts already approached in the topic of Data Preprocessing (section 2.3). In fact, Feature Engineering can be seen as a step of the KDD process posterior to DP or as part of it, taking into account both activities transform and redesign data in order for the ultimate analysis (in this case, anomaly detection) to be more successful.

There are several FE methods. Bellow follows a description of the most commonly used ones. Though there are others, the methods detailed here are those that are potentially more useful for the project of this dissertation.

2.4.1 Statistical Features

Often very helpful to understand time series behaviour (Caesarendra and Tjahjowidodo, 2017), features of this type are very common and simply correspond to statistical measures of data. The most common examples inlcude the mean, standard deviation and variance. More advanced features include skewness and kurtosis, which analyse the probability density function (PDF) of the series.

Zero Crossing (ZC) and Slope Sign Change (SSC) are other examples of statistical features that can be extracted in the time domain (Altin and Er, 2016). The first corresponds to the count of times that amplitude values of the signal cross the zero-amplitude level. The latter corresponds to the count of the number of times that the slope of the signal changes.

2.4.2 Principal components analysis (PCA)

Principal Component Analysis is a powerful statistical-based method that can obtain new features from the original ones. It allows the conversion of multiple variables to a few comprehensive features that can reflect most of the information of the original variables. This conversion is obtained through linear transformation and the obtained variables are called principal components. These principal components are both linearly independent of each other as well as mutually orthogonal (Men et al., 2018). In more basic terms, the PCA linearly transforms the data so that the cross correlations between features is zero (decorrelated). The principal components compose the transformation matrix.

This is a very popular technique, used across many areas though it is only suited for numerical data. When applied, input data is firstly normalized so that each attribute falls within the same range, preventing one feature of dominating the analysis. Then, k orthogonal vectors are computed, considering that the input variables are linear combinations of these k principal components. The principal components are also sorted in order of decreasing strength, allowing the elimination of weak components (Yang, 2013).

2.5 Anomalies

Before defining Anomaly Detection, one must previously define the concept of anomaly. In a data science perspective, anomalies are defined as patterns in data which do not resonate with a defined notion of normal behaviour. Such atypical behaviour is of the interest of the analyst, and itself is the most relevant feature for anomaly detection (Agrawal and Agrawal, 2015).

It is also important to note that anomalies are not quite the same thing as outliers. Rather, they are more of a specific type of outlier. Anomalies include unexpected or rare pattern occurrences.

As referred by Cook et al. (2020) and Liu et al. (2020), early literature describes outliers as observations that deviate significantly from other observations in the same data set, arousing suspicions of having been generated by a different mechanism. Another way to understand outliers is as observations (or subsets of observations) that are very inconsistent with other sample points in their related data set.

Thus, as reported by Blázquez-García et al. (2020), the difference between a common outlier and an anomaly resides in the fact that an anomaly, once detected, is kept by the analyst for further investigation. Common outliers, on their hand, are described as values that are not generated by the main process and thus, are discarded with the intent of cleaning the data set. This way, general outliers constitute unwanted data, which are supposedly generated by processes external to the functioning of the system data generates the data (software errors for example). Anomalies are a species of outlier that is of interest to the analyst, i.e., that form events of interest. Contrary to common outliers, which, once found, are discarded from the data, anomalies are kept and investigated.

A variety of terms have been used in the literature to refer to anomaly. These unexpected behaviours are frequently termed as outliers, discordant observations, among others. However, the term outlier is clearly the most used substitute in the context of anomaly detection (Agrawal and Agrawal, 2015; Chandola et al., 2009). This way, once the two concepts significantly overlap, the generality of the anomaly detection techniques here exposed are considered outlier detection techniques as well.

2.5.1 Types of anomalies

The choice of which AD technique to use is also dependent on the kind of anomaly we look for. As shown in Figure 2.3, anomalies are classified in three categories, depending on their nature: point anomalies, subsequence or collective anomalies and anomalous time series (Chandola et al., 2009). Naturally, each AD technique is better suited for some types of anomalies than it is for others.

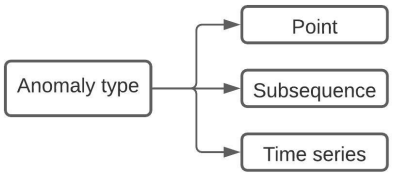


Figure 2.3: Types of Anomalies

Point anomalies: according to Blázquez-García et al. (2020) an anomaly of this type corresponds to a data point that behaves abnormally when compared to other data points. This definition is very similar to the one previously given for a general anomaly, and in fact, this is the type that is traditionally most found (Cook et al., 2020). It is the simplest type of anomaly and the focus of the

majority of the research as well (Chandola et al., 2009). It should also be noted that point anomalies may be multivariate, in case of affecting a plurality of time-dependent variables in a MTS, or univariate in case the TS is univariate or if it affects only one variable in a MTS (Blázquez-García et al., 2020).

Subsequence/collecctive anomalies: a subsequence anomaly is represented by a group of consecutive data points whose joint behaviour and appearance in a particular form is anomalous with respect to the rest of the data. Individual observations of these groups of points may or may not be an anomaly by itself, i.e. a point outlier (Cook et al., 2020). Regardless of that, it is their occurrence together that defines the group as a collective anomaly. In this way, techniques for detecting collective anomalies present several differences with respect to point anomaly detection techniques (Chandola et al., 2009).

Similarly, to point anomalies, collective anomalies can also affect one or more time-dependent variables, thus being univariate or multivariate anomalies, respectively (Blázquez-García et al., 2020). Additionally, detecting these anomalies is more challenging than detecting point outliers, involving the exploration of the several structures of the data in the anomalous regions (Chandola et al., 2007). Subsequence anomalies can only occur when there is a relationship between data instances (Chandola et al., 2009).

Anomalous time series: this is the less emphasized type of anomaly in the literature. It can only be found in multivariate data and corresponds to the case where an entire time-dependent variable of the TS behaves unusually, hence the name (Blázquez-García et al., 2020). This kind of anomaly is not approached at all in this dissertation.

2.5.2 Global and local anomalies

One last issue regarding the characteristics of anomalies has to do with how much data is a potential anomaly is compared to. To be classified as an anomaly, an observation or sequence of observations must behave unusually (be different in most of its aspects) when compared to the rest of the data. However, frequently, not all the data of a time series is used to make this comparison. Instead, a portion of data which includes the potential anomaly is used for that task. This gives place to two types of anomalies: global and local.

Global anomalies are those whose behaviour is anomalous with respect to the entirety of the time series. In contrast, if the anomaly detection technique only analyses a portion of the data, then any potential anomalies found there are considered local anomalies, precisely because they were looked for in a specific location of the data. Global anomalies can also be local, but the contrary is not always true. In other words, local outliers may appear normal when observing the complete data set but may be anomalous when analysed in their neighbourhood (Blázquez-García et al., 2020). This way, for two observations with the same behavioural attributes (for example two points representing two corks with the same TCA concentration), one may be an anomaly due to its contextual attributes, and the other a normal point in a different context.

2.6 Anomaly detection

According to Cook et al. (2020), anomaly detection can be described as the identification of patterns which do not conform to the expected norm of the system. This description comes into agreement with the previous concept of anomaly, showing that AD is the act of finding data patterns that do not conform to its expected behaviour, i.e. anomalies.

Taking this into account, the importance of anomaly detection is easily understood. Anomalies in data carry significant information, often critical, for the comprehension of the functioning (and malfunctioning) of a system in a wide variety of application domains. Thus, detecting and analysing them is an important task. Examples of AD domains are fraud detection for credit cards, health care or fault detection in automated systems (Chandola et al., 2009). Hence, when data needs to be analysed for purposes such as relationship identification or data prediction, anomaly detection may come into action using its various techniques.

2.6.1 Types of anomaly detection techniques

The output of an anomaly detection technique is the identification of a data point (or subsequence) as normal or anomalous. This is done through the creation of a variable that discriminates each case. For example, adding a variable "anomaly" to the data, the algorithm would fill the value of this variable for each observation identifying it as normal data or as an anomaly (or as a specific type of anomaly).

As already stated, the input data used for training an algorithm (training data) plays an important role in selecting the appropriate ML technique. Input data may present a variable that labels data as normal or anomalous before using the algorithm. This is known as ground-truth label. Therefore, the algorithm would then try to approximate its results to those provided in the ground-truth label.

However, obtaining labelled data is often too expensive and resource consuming because it is usually done manually by a human expert. Furthermore, it is only worthy to do this if the labels are accurate and representative of both normal and anomalous data (including all types of anomalies) which frequently is also extremely difficult. It is hard to account for all types of anomalies without previously employing an anomaly detection algorithm, leaving space for unrepresented anomalies. Furthermore, in dynamic processes, data changes may create new anomalies, previously inexistent, and eliminate others that had been identified (Chandola et al., 2009).

Taking this into account, AD techniques generally fall into one of three categories: supervised, unsupervised, and semi-supervised. This classification is based on the style of learning and the availability of ground-truth labels (Shanthamallu et al., 2018).

2.6.1.1 Supervised anomaly detection

This kind of anomaly detection represents algorithms that work with labelled data (use of ground-truth labels in the model training). During the training process, the model tries to predict

whether data is anomalous or not in such a way that it approximates to the labels already presented (Shanthamallu et al., 2018). There is also the assumption that normal data and all types of anomalies are well represented in the input data. If this is not verified, data may be imbalanced, leading to biased models that may be accurate classifying data of the training data set, but may perform poorly in the presence of new data (Chandola et al., 2009).

The algorithm is built to classify or predict the label of the training data as accurately as possible, but it should also be accurate in the presence of new data (Das et al., 2017). This can be verified using the algorithm in another set of labelled data, called test data. If the accuracy remains high, it means the algorithm is unlikely to be biased or overfitted to the training data.

2.6.1.2 Semi-supervised anomaly detection

In semi-supervised learning, the assumption is that the training dataset presents some labelled instances, both normal and anomalous (that is, the dataset is only partially labelled). However, it is more common to only have normal labels. Therefore, semi-supervised techniques are usually more widely applicable than supervised techniques, not requiring the difficult task of collecting labelled anomalous data (Chandola et al., 2009). Nonetheless, though rarely, some of these techniques also use labelled anomalous data, using a combination of labelled and unlabelled data for training (Das et al., 2017).

2.6.1.3 Unsupervised anomaly detection

Unsupervised ML on its hand does not make use of any labelled data (Das et al., 2017), which make these techniques more widely applicable. With these methods there is the assumption data normal observations are more frequent than anomalies. This way, if an observation presents characteristics that are uncommon, then they probably represent an anomaly. If this assumption is not verified, then the algorithm will exhibit a high false alarm rate (Chandola et al., 2009). Algorithms in unsupervised techniques draw inferences from the data and then model the underlying structure and distribution of the data (Shanthamallu et al., 2018). This means that they are designed to deduce patterns or clusters in the datasets (Das et al., 2017).

2.6.2 Output of anomaly detection techniques

A final note two the way AD techniques report its findings. Typically, there are two main ways in which anomalies are reported: using a score or a label (Cook et al., 2020).

Score-based techniques report anomalies using a score that denotes the degree of anomalous behaviour each anomaly presents (Cook et al., 2020). As such, the output of these methods can be seen as a ranking of anomalies which allows analysts to use domain-specific thresholds and prioritize deeper analysis in anomalies with higher scores (Chandola et al., 2009).

Other techniques use labels to report anomalies. A label (normal or anomalous) is assigned to each observation (or group of observations) noting whether the algorithm recognized it as an anomaly or not. Though these techniques do not allow analysts to directly differentiate between

anomalies, that can be achieved indirectly through parameter choices within each technique (Chandola et al., 2009). This approach is mostly used when the identification of anomalies must be done in near real-time notification of unusual states. More complex methods, may use multiple label classes instead of a binary system – *normal/anomalous*. This way, different types of anomalies can be identified with different labels – *normal*; *anomaly A*; *anomaly B*, etc. (Cook et al., 2020).

2.7 Anomaly detection techniques

Contemplating the information so far reported in this chapter, some considerations must be done regarding the anomaly detection techniques that follow bellow. Firstly, the data under analysis constitutes a Multivariate Time Series and so, the techniques here present are focused on detecting anomalies in this type of data.

Secondly, as the title of this document suggests, the focus of this project is to find anomalous patterns. This meets the idea of detecting anomalies of the collective type. For this reason, it could be assumed that only techniques that detect subsequence anomalies are analysed. However, there are some methods which enable the transformation of anomalous subsequence detection into point anomaly detection. This is primarily achieved through Feature Engineering. This way, techniques to detect both groups of anomalies are reviewed. Anomalous time series are not approached here, once there is no consideration of the possibility of an entire variable being anomalous.

Another aspect that must be mentioned concerns contextual anomalies. Given the fact that the observations studied in this project maintain a relationship between in each other under the form of temporal order, it makes sense to analyse anomalies considering its context (Chandola et al., 2009). The usual approach is to divide the time series into equal-length subsequences and then look for anomalies in each of them separately. A defined length for a window of analysis is defined and then data is evaluated as this window sequentially moves along the data. This is commonly known as sliding window. Though it is computationally expensive, it is adequate for cases with dynamic data where a normal observation may be considered anomalous in a different context such as the case with NDtech's data (Liu et al., 2020). Regardless, the AD techniques themselves can be applied over these sliding windows or over the entire time series. For this reason, this topic is not approached bellow, all presented techniques can be used to detect global or local anomalies. In Chapter 4 however, every time an approach is defined and used this issue will be noted.

Furthermore, the initial steps of this project focus on cataloguing the anomalies present in the data. In other words, the anomalies will be identified and classified into a group, without prior information regarding the behaviour of the data, i.e. without ground truth-labels. This way, the first approaches will consist of methods under the spectrum of unsupervised learning. After this, supervised learning techniques will be employed to identify anomalies in new data. For this reason, unsupervised techniques will be presented first and then the supervised ones.

Lastly, a brief explanation behind the reasoning for the order of appearance of the techniques. As referred, all techniques reviewed are used for detecting anomalies in MTS. First, unsupervised techniques are presented before supervised ones. Within each of these, groups point anomaly

detection techniques are the ones to be presented first followed by subsequence anomaly detection methods.

2.7.1 Unsupervised techniques

In the context of this project, clustering is the most suited unsupervised machine learning technique for anomaly detection. Though there are other approaches that could also be considered, such as Vector Quantization, Spectral techniques, or Information-theory based techniques, these do not fit so well to the profile of this project. For that reason, this review focuses solely on the clustering approach.

2.7.1.1 Clustering

Clustering algorithms divide the dataset into different groups/clusters of similar objects by projecting the data into a multidimensional space (Cook et al., 2020). As a result, data points are similar to other data points within the same cluster and dissimilar to data points in other clusters (Shanthamallu et al., 2018). Observations within dense clusters are indicated as normal observations while points that do not belong to these clusters and are far from them are considered anomalous (Cook et al., 2020). This technique finds anomalies in data without prior knowledge, being primarily an unsupervised technique, though semi-supervised clustering has also been explored (Chandola et al., 2009).

Different distance metrics (Euclidean, Cosine, Minkowski) can be used to assign an observation to a cluster and to verify the quality of that assignment. Euclidean distance is the most used metric, though unsuitable for capturing the quality of the clustering (Shanthamallu et al., 2018).

As denoted by Agrawal and Agrawal (2015), there are several methods to perform clustering-based anomaly detection:

- ***K-means*** is one of the simplest clustering algorithms. It iteratively clusters data by separating it into k disjoint groups on the basis of each point's features (Agrawal and Agrawal, 2015). K is a user defined parameter, being determined by the analyst prior to the application of the algorithm. Each observation is assigned to the cluster with the nearest mean, $\mu_{(k)}$, also referred to as the *Centroid* of that cluster. Thus, each cluster is defined by its centroid (Shanthamallu et al., 2018).

- ***K-medoids*** differs from *k-means* in the representation of the different clusters. With this method, each cluster is represented by the most centric object in the cluster, instead of its implicit mean (centroid) that may not belong to the cluster. This approach is more robust than *k-means* because a *medoid* is less influenced by outliers or other extreme values than a mean, often producing results with a much higher accuracy (Agrawal and Agrawal, 2015).

- ***Expectation-Maximization clustering*** is another clustering algorithm. *EM clustering* can be seen as an extension of *k-means* which assigns an object to the cluster based on the mean of the clusters. However, in this approach, the assignment of each object to its cluster is

based on a weight representing the probability of membership. This creates strict boundaries between clusters. *EM clustering* usually outperforms *k-means* and *k-medoids*, obtaining higher accuracy (Agrawal and Agrawal, 2015).

These algorithms can then be employed in three different ways to detect anomalies (Chandola et al., 2009). The first way is assuming that normal data points belong to a cluster in the data while anomalies do not belong to any cluster. Thus, points not belonging to any clusters are declared anomalies by the algorithm. The disadvantage of performing clustering with this assumption is that these algorithms are not optimized for anomaly detection, rather they are constructed with finding the clusters as its main goal.

The second way assumes that normal observations are close to their closest centroid (or medoid), while anomalies are far from it. This approach uses two steps, clustering the data first and calculating each observation's distance to its closest cluster centroid as its anomaly score afterwards. An example of this is Self-Organizing Maps (SOM), proposed by (Smith et al., 2002).

In case anomalies in the data form their own clusters, the techniques based on the aforementioned assumptions will not detect them. To overcome this problem, a third way of performing clustering-based anomaly detection is based in the assumption that normal data points belong to large and dense clusters, while anomalous observations are alone or belong to small or sparse clusters. This way, observations in clusters whose size and/or density is below a predefined threshold are declared anomalous. An example of such technique is CBLOF (Cluster-Based Local Outlier Factor), proposed by (He et al., 2003).

One of the most utilized clustering algorithm is DBSCAN. This algorithm constructs clusters based on the density of the neighbourhood of each point. This way, points who have a very dense neighbourhood, that is, with a minimum defined number of points within a certain radius will be considered core points. Points within that same radius will form a cluster and points not in the neighbourhood of any other point are considered anomalies. With this, several clusters are created. The total number of clusters created is not defined by the user, which is an advantage of the algorithm.

With this in mind, (Chandola et al., 2009) noted some additional characteristics of clustering anomaly detection. The testing phase can be considered fast, involving only the comparison of the data to a small number of clusters' centroids (or medoids). Furthermore, the technique can be easily adapted to other complex data types by using a clustering algorithm capable of handling the particular data type. On the other hand, several algorithms of this species force every observation to be assigned to a cluster. This may result in anomalies being assigned to a large cluster, thus being later misclassified as normal data points if the technique assumes that anomalies do not belong to any cluster.

2.7.2 Supervised techniques

In the case of supervised approaches, classification techniques are those that clearly have the most potential in the context of this project. Therefore, the algorithms that follow can all be

considered classification algorithms.

2.7.2.1 Classification

In general, classification is defined as a question of identifying the class of new observations using the information provided by a training dataset which contains previously classified instances, which meets the characteristic of a supervised learning model (Agrawal and Agrawal, 2015). Therefore, a classifier learns the intrinsic structure of the data in a first phase using labelled observations and then classifies new observations using that information in a second phase.

The algorithm that implements the classification, i.e. the classifier, is used to predict categorical attributes. In the anomaly detection domain, the prediction will assign each point as normal or anomalous. Such is the case for a basic one-class classification problem, that only distinguishes between anomalous and normal data. In a multi-class classification problem, the algorithm may further distinguish between types of anomalies or between types of normal observations. This will of course depend on the available labels provided in the training phase of the algorithm (Chandola et al., 2009).

Next, a variety of classification algorithms used for anomaly detection are provided. Besides the ones here presented, Association Rules, Linear Regression, Isolation Forest and Local Outlier Factor are other interesting supervised techniques that can be used.

Naive Bayes:

Bayesian algorithms are simple probabilistic classifiers based on the Bayes theorem. This model outputs the posterior probability of each observation belonging to a class considering its input features (Cook et al., 2020). The probabilities used are inferred from the training dataset and then used to classify new instances (Shanthamallu et al., 2018).

Naive Bayes assumes independence between the different attributes, which is the reason for the use of the term “Naive”. However, this is frequently not true, existing some degree of correlation between some or all variables. For that reason, some variations of this model have already been proposed to tackle this issue (Chandola et al., 2009).Compared to other classification algorithms, Bayesian techniques are often less accurate. However, they present lower computational time and thus are preferably used when data is very large (Agrawal and Agrawal, 2015).

Decision Trees:

Also termed as classification trees, this method uses a tree pattern graph similar to a flow chart structure where its internal nodes are a test property and each branch represents a possible test result. Final nodes, or leaves, are the classes to which an object may be assigned to. Decision trees can be constructed in two fashions, top-down or bottom-up pruning. The most fundamental algorithms used are *ID3* and *C4.* and both are examples of top-down tree construction (Agrawal and Agrawal, 2015).Compared to other approaches, namely Naïve-Bayes classification, decision trees are often more accurate.

Support vector machine (SVM):

SVMs have been mostly used for binary classification, i.e., for one-class problems, distinguishing only between normal and anomalous observations and not between types of anomalies

(Chandola et al., 2009). In such setting, this technique is used to map the input data in an 'n' dimensional space and draws an '$n-1$' dimensional hyperplane to separate data points in two groups. This way, using training data, the algorithm learns a region that contains normal instances, defining decision boundaries between normal and anomalous data such that the width of separation between the two groups is maximized (Shanthamallu et al., 2018). The data points close to the defined hyperplane, i.e. the boundary, are called support vectors. Then, to classify each new instance, if it falls within the normal data learned region, it is deemed as normal, otherwise it is declared anomalous. For multi-class problems, where multiple anomaly classes exist, the algorithm learns multiple regions, using multiple hyperplanes. Each data instance is then classified in accordance to which region it falls in. This is similar to the procedure seen in the unsupervised learning technique of clustering.

Artificial neural networks:

As the name suggests, Artificial Neural Networks (ANN) are designed to imitate the human brain using a set of interconnected nodes. The peculiarities of this technique have made some consider it a different branch of Machine Learning, belonging to Deep Learning, a name that derives from the use of multiple consecutive layers to learn different levels of representation (Shanthamallu et al., 2018). In fact, an ANN's structure consists of multiple layers which contain several nodes. Each of these nodes has a weighted connection to several others in neighbouring layers. Taking the outputs from previous nodes, and using the weights in a simple function each node computes its output and forwards it the next ones (Agrawal and Agrawal, 2015).

Different architectures can be constructed using this technique, with the user being able to define the number of layers, the number of nodes in each of them and the way they are connected, that is, the topology (Shanthamallu et al., 2018). This enables ANN to be used both in supervised and unsupervised learning environments, though the most typical and basic use is in a multi-class supervised learning. This is the principle of Autoencoders. The training step involves compressing data into the hidden layers. Then, when testing, the model tries to reconstruct each data instance using the learned network to obtain the reconstructed output. The reconstruction error then used to determine an anomaly score for the instance, thus classifying it as normal or anomalous (Chandola et al., 2009).

Classic challenges for this model include overfitting, long training time and initialization of the network parameters. Nowadays, some strategies have been developed to overcome these difficulties, such as data normalization which reduces training time. In the process of evolution of this technique to a more robust model, some variations have been created, with the most known examples being Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) (Shanthamallu et al., 2018).

Logistic regression:

Logistic regression is a binary classification technique that presents a discrete output. It can be described as a statistical way of modelling a binomial outcome. This way, the model assigns each observation a value of '0' or '1', depending if it is considered normal or anomalous This technique uses a sigmoid function to calculate the probability of a record belonging to either class, '0' or '1'.

Thus, the output of the sigmoid function outputs a value between 0 and 1. If that value is below a defined threshold, for example 0.5, then it is assigned to the negative class and. If it is greater than said threshold than it is assigned to the positive class (Shanthamallu et al., 2018).

Nearest neighbours:

Techniques based on the Nearest Neighbour (NN) concept generally assume that normal data points occur in dense neighbourhoods while anomalies occur far from their closest neighbours, presenting themselves as isolated points. Most applications use NN based models to classify input data to discrete outcomes, which is done through the means of a similarity measure, i.e. a distance metric. According to the input data type, this measure can be calculated differently. For continuous attributes, Euclidean distance is the most common choice while for categorical variables the usual choice falls over a matching coefficient (Chandola et al., 2009).

As Chandola et al. (2009) noted, an important detail regarding NN AD techniques is that these can be subdivided in two groups. First, there are the k-NN models, which use the distance of an instance to its k^{th} nearest neighbour as an anomaly score. Usually, this score is then compared against a predefined threshold to determine if the instance is an anomaly or not. The other group of techniques based on the NN concept are those which calculate the relative density of the neighbourhood of each data instance to compute its anomaly score. These are the density-based models. Points in a low-density neighbourhood are classified as anomalous and points in a dense neighbourhood are deemed as normal.

Being a non-probabilistic and non-parametric model, it is the first choice for many classification studies when there is no prior knowledge about the data distribution. Though models constructed on this concept present some of the simplest ML algorithms, they are very computationally expensive since all instances have to be stored in memory in order to compute all distances between each pair of records. This complexity may even be aggravated if data is multidimensional, reason why dimensionality reduction techniques are often employed before using this method (Shanthamallu et al., 2018). Other strategies have been developed to increase model's speed such as sampling,though this might result in incorrect anomaly scores if the sample size is too small (Chandola et al., 2009).

There are clear similarities between Nearest Neighbour techniques and Clustering techniques. They both depend on calculating distances between each two points, and thus suffer from the same issues regarding that aspect (computational complexity). However, besides the first being primarily applied in supervised settings and the latter in unsupervised ones, the key difference lies in the fact that clustering evaluates each instance with respect to which cluster it belongs to, measuring the distance of the observation to the cluster's centroid (or medoid). Nearest Neighbour techniques on their hand, analyse each point with respect to its local neighbourhood.

Pattern matching:

Pattern matching is a method that directly models the time series. In supervised settings, knowing the characteristics of anomalous subsequences, the detector compares each new instance against a database of labelled anomalies, flagging those which are more similar. In unsupervised

settings, the detector learns the most common historic patterns of the normal data and signals as anomalies those new sequences that do not match the historic patterns.

Statistical techniques:

All statistical AD techniques assume that an anomaly is an observation suspected of being partly or fully irrelevant due to not being generated by the stochastic model assumed. Normal data points occur in high probability regions of that model, while anomalies occur in the low probability regions (Chandola et al., 2009). This way, such techniques use historical data to create a statistical model that captures the expected behaviour of the system under analysis. Consequently, when a new test observation arrives it is compared against the model and if it does not fit that model, it is considered an anomaly (Cook et al., 2020). Such comparison is performed using a statistical inference test, which produces a test statistic, used to classify the observation. Using a boxplot or a 3σ approach to identify anomalous points is very common. The main advantage of using statistical techniques is the ability to provide a statistically justifiable solution for anomaly detection if the data distribution's assumptions are correct. On the other side, the principal disadvantage is the reliance on the assumption that data is generated from a particular distribution, which often is not true, especially for high dimensional data sets.

Chandola et al. (2009) divide the category of statistical techniques in parametric and non-parametric techniques. The first kind assumes that the underlying distribution is known and estimate its parameters from the data to create the model. According to the assumed distribution, parametric techniques can be Gaussian or Regression, for example. Non-parametric techniques, however, do not assume that the distribution is known (Chandola et al., 2009). This means that the model's structure is not defined *a priori*, but instead is determined from the given data. The simplest example of such non-parametric statistical techniques is using histograms to obtain a profile for normal data, and then checking whether new observations fall in any of the histogram's bins. If so, then it is a normal instance, otherwise it is an anomaly.

2.8 Sliding Windows

Because large quantities of data are produced and provided for analysis, it may be too expensive to use the entire available dataset for analysis, especially, when the analysis is performed on resourced constrained machines. With this in mind, windowing techniques have been developed (Cook et al., 2020). In short, a sliding window approach reduces the memory and storage requirements for the processing system, allowing for an analysis in a short collection of observation within a sequence. As such, this method processes a subsequence within a larger sequence of data as a unit element for analysis (which later allows the use of point anomaly detection techniques). This is not an anomaly detection method itself, but rather an addition to such methods that helps in the discovery of the anomalies.

Window-based strategies extract fixed-length overlapping windows from a sequence of data. Each of these windows is then treated as a point that will be classified as normal or anomalous. This is particularly interesting when looking for anomalous subsequences of observations and

the anomaly can be traced to one or a few shorter substrings within a large sequence. If, on the contrary, the entire sequence is analysed, the anomaly might not be distinguishable. This way, with this procedure, it is possible to localize an anomaly within one or a few short windows (Chandola et al., 2012). The classic procedure to obtain short windows from a sequence of data is to slide a fixed-length window along the data, one instance at a time. This way, each window will only be deviated by one observation from the previous, with the rest of the instances being the same, hence the term 'overlapping' windows. Choosing the optimal length of the window is critical.

2.9 Comparison of techniques

As a final approach to all described techniques here follows a succinct comparison between the generality of them and their respective merits and disadvantages.

When data is simple, with one or just a few dimensions, with anomalies presenting very distinct characteristics from normal data and these being very similar between each other, virtually all above-mentioned techniques can be applied for anomaly detection. In such scenario none of them would have great difficulty detecting anomalies, though, depending on the specific conditions, some may be more suitable than others.

However, when data sets are more complex, different techniques face different challenges. Nearest Neighbour and Clustering based techniques are more challenged by high dimensional data because measuring distance in a large number of dimensions is notably expensive and segregation between normal and anomalous data is hampered. Furthermore, these two groups of approaches need proper identification of a good distance measure, which sometimes is hard to do. For this reason, other classification-based techniques may be a better choice in some cases.

Spectral techniques are especially designed to address the issue of high dimensionality by mapping data to a lower dimensional projection. On the other hand, these techniques are highly dependent on the assumption that normal and anomalous instances are differentiable in the projected space, which in certain cases is not true for many anomalies.

A Classification-based technique would be a better choice in the general case. Nonetheless, to be highly effective, these approaches require labels for normal and anomalous instances, which often are not available. Besides, even if they were, the imbalance between the distribution of the two label is a problem to consider and it can lead to a more challenging learning task. Semi-supervised NN or Clustering techniques that use only normal data labels are in most cases more effective than many classification-based techniques.

Statistical techniques, even if used in an unsupervised fashion are effective only if data is dimensionally small and statistical assumptions are true. Information Theoretic methods, on their side, need a measure sufficiently sensitive to detect the effects of just a single anomaly or a very small number of anomalies. Otherwise, these techniques can only detect anomalies if they are present in a somewhat large number.

Finally, a note should be given to the fact that the generality of anomaly detection techniques assume that anomalies are rare when compared to normal data points, and though this is generally

true, there might be cases where it is not. In such cases, where anomalies are present in a great number compared to normal data, unsupervised techniques are not indicated. Supervised and semi supervised techniques work better for such bulk detection scenario.

Chapter 3

Problem description

In this chapter a detailed presentation of the process taking place at NDtech is given. This includes all the steps that constitute the process, its specific set-up in NDtech and the control methods implemented by the company to obtain the best results. Following this, the most common issues regarding the equipment are approached. Furthermore, some considerations are made about the data available to analysis, which are complemented with a description of how the company is currently using that data. After this, it is discussed how detecting and identifying anomalous patterns in the data constitutes an interest and an advantage to the company in order to better control the process, obtaining better results for the organization. Finally, taking all this information into account, the work proposal is defined, encompassing the main objectives of the project.

3.1 The process of detecting TCA

As mentioned, this work revolves around the technology that Amorim Cork developed to detect the presence of TCA in its cork stoppers. NDtech is the name of that technology and also the name of the department in charge of running and operating everything related to it.

The method used to perform that detection is Gas Chromatography – Electron Capture Detection, a chemical analysis process. Amorim took this technology and designed each detection unit to perform a TCA oriented chromatographic analysis. A scientific explanation of what Gas Chromatography is, and how it is used at NDtech follows bellow.

3.1.1 Gas chromatography

Extremely low concentrations of TCA such as 1.5 ng/l are already enough to diminish a wine's quality due to its effect on the wine's odour and taste (Sefton and Simpson, 2005). One way to make sure that cork stoppers for wine bottling do not exceed acceptable levels of TCA is using Gas Chromatography to identify and reject corks with exceeding TCA levels.

Chromatography is the most commonly used procedure in contemporary chemical analysis (Scott, 2003). The gas chromatographer was one of the first analytical instruments to be commercially available (1955) and associated with a computer which controlled the analysis, processed the

data, and reported the results. Gas chromatography fundamentally corresponds to the separation of different compounds carried by a gas flow, by transporting such flow through a chromatographic column. The column is the component responsible for the segregation of the different molecules, allowing for the individual recognition and quantification of each of them.

Scott (2003) defined the typical gas chromatographer as being constituted by four main interconnected parts: the gas supply, that provides the necessary carrier gas; the sampling unit, where the sample is placed; the column unit, where the chromatographic column performs the actual separation of the mixture's components; and the detector, which detects the components that separately arrive at the end of the column. A scheme of the general configuration of chromatographer is represented bellow on figure 3.1.

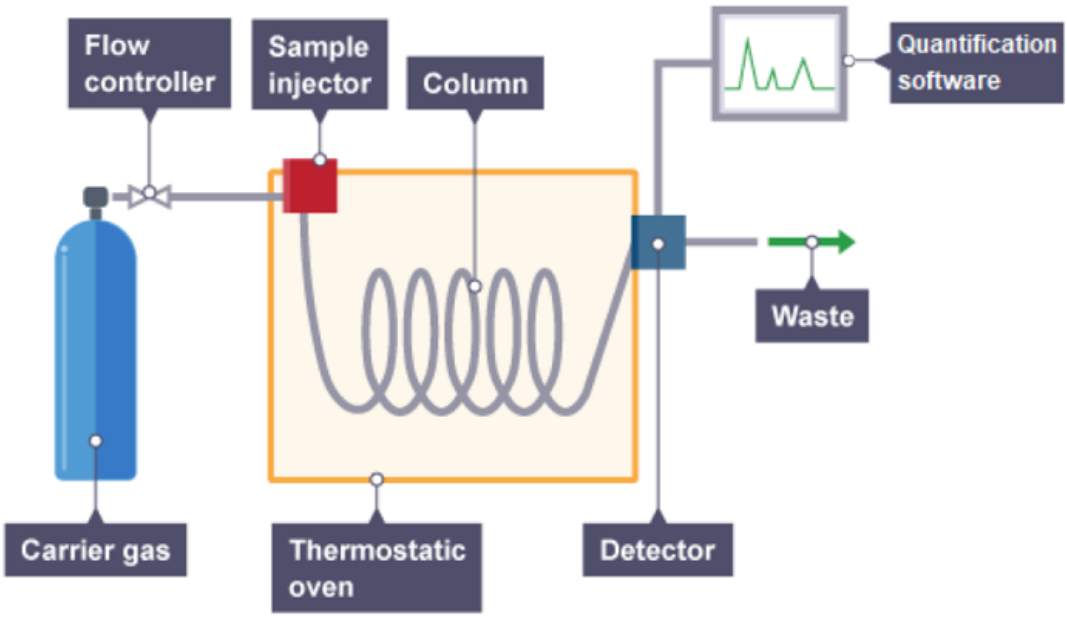


Figure 3.1: General gas chromatography configuration (adapted from Delloyd's Lab-Tech Chemistry (nd))

All these parts of the instrument work together and sequentially. The sampling unit is a thermostatically controlled enclosure/chamber that receives the sample and then, carrier gas coming from the gas supply unit is injected inside it as well. Once the carrier gas starts being injected in the sample injector, the flow is mixed with the compounds present in the sample and the mixture is carried to the next part of the chromatographer, the column. That is where the actual separation of the compounds occurs. When the mobile phase, i.e., the gas, passes through the column different molecules will travel at different speeds depending on factors such as their molecular mass and volume or their interactions with the stationary phase, which is a special internal coating of the column. This way, the analyte's compounds are segregated, reaching the end of the column separately and striking the detector at different times. There, at the detector, the quantity of each compound is measured and data is sent to the processing computer.

There are other details that should be mentioned as well regarding the chromatographic process. First, it is important to note that the carrier gas is usually inert so that it minimizes the interactions with the sample gas mixture. Secondly, it should be acknowledged that there are many types of chromatographic columns commercially available, varying in its dimensions, materials, and internal coatings. Usually, this last one is chosen based on the polarity of the analyte one wants to study. For example, TCA is an apolar molecule, and thus, an apolar coating is also the best choice (Harris, 2008).

Also, it is essential to control the pressures and flows of the carrier gas and of the mobile phase to ensure correct measurements. Temperature is another important variable to be controlled. Ovens are used to raise the temperature along the whole path of the process to facilitate the flow of the mobile phase and to ensure the analyte remains in the gas phase (Harris, 2008).

Finally, there are several types of detectors. Electron Capture Detector is one of the most used detectors, and the one used at NDtech. In essence, this kind of detector possesses an electron emitter and an electron receiver. A small potential (few Volts) is applied to the receiver, enough to capture the emitted electrons and create a standing stable electric current. This way, when an electron capturing molecule enters the detector, electrons are captured, and these molecules become charged. In the case of TCA, the chlorine atoms which are electronegative are responsible for capturing the electrons. Fewer electrons will then reach the receiver, causing a drop in the previously constant current, in accordance to the quantity of molecules arriving to the detector. This drop in the current is what allows the quantification of the chemical compounds, such as TCA, in a sample. The readings of the detector peak when a certain molecule is at its maximum density. This peak time is empirically studied and used to identify the different compounds present in the analyte. This way, Amorim determined at what time do the TCA molecules arrive to the detector, considering all factors involved, such as length of the column, temperatures and pressures along the path, etc. The area beneath the signal around the region of the peak indicates the concentration of TCA in the analyte..

3.1.2 The GC-ECD set-up at NDtech

The set-up of the chromatography units at NDtech corresponds to Gas Chromatography – Electron Capture Detector (GC-ECD), which functioning was presented above. There are 60 detection gas chromatography units (also termed as modules) that perform the individual analysis of each cork, indicating the concentration of TCA present on their surface. The GC units are equally distributed along 10 lines (or machines). Each line has its own automatic cork distributing system, where stoppers are manually supplied in batch by an employee. This way all modules in a line share the same cork feeding system, leading them to analyse corks from the same lot at the same time.

Besides having several chromatographic units performing analyses at the same time and continuously, the company also designed each module in order to maximize their throughput. First, before being inserted in the module's sampling chamber, stoppers are preheated in a separated oven (the pre-incubator), raising their surface temperature to 70 ºC, which facilitates the release

of TCA from the corks' surface. Additionally, each module has eight sampling units, or chambers, which are thermostatically controlled as well, and where corks are placed. For this reason, each module can receive eight stoppers simultaneously. The unit has also four chromatographic columns each shared by two of the chambers, but only one detector, shared by all columns. The system is designed so that valves control the flow of each of the 8 samples ensuring each flow reaches the detector alternately. This way, only one sample is analysed at a time in the detector.

Indeed, each module takes two minutes to analyse the concentration of TCA on a single stopper, but only 15 seconds correspond to the actual measurement in the detector. Using the set of valves, and a 15 second lag between the placement of a cork in each chamber, the flows of the eight chambers are directed to the appropriate path inside the unit. As such, along the two-minute cycle, though only one flow is being analysed, the other seven are already in different stages of the whole process, such as analyte separation in the columns, or purge of the sample after flowing through the detector. This results in a cork being analysed every 15 seconds in a single chromatographic unit. Thus, it should be noted that one measurement cycle in the modules corresponds to two minutes, that are divided in eight different analyses, each taking 15 seconds.

The following details should also be considered. A sample is extracted from a cork using the injection of the carrier gas in the chamber, which drags the compounds in the cork's surface, carrying them to the corresponding column. When the mixture reaches the detector, the concentration of TCA in the sample is calculated by integrating the area of the signal below the peak of TCA. A software is employed to perform the calculations, as well as to define the system's parameters and to present a graph with the curve of the signal and the final classification result of the stopper. In reality, NDtech uses two different software systems to calculate the integral. The final and correct concentration, however, is assumed to be the highest of the calculated values by the two softwares, thereby adopting a conservative approach to the estimation of the TCA concentration. For each chamber, a cycle begins with the insertion of a cork inside it and ends when the cork is removed from it to be conducted to a basket according to its TCA estimated concentration.

Figure 3.2 shows the main steps of the measurement cycle occurring in the chromatographic units:

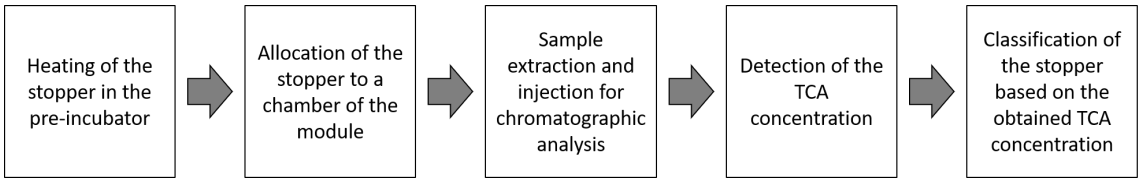


Figure 3.2: Main steps of the TCA detection cycle in NDtech's modules

The TCA measurement is compared against predefined thresholds and then categorized in one of three classes. If the area for the signal is lower than 6 mV.s than it belongs to Group 1 (G1). If the area is between 6 and 25 mV.s then it is a Group 2 (G2) cork. If it presents an area corresponding to more than 25mV.s then it is considered a Group 3 (G3) cork. Only Group 1 stoppers are considered to have TCA levels low enough nor to be undetectable by humans nor to

taint wines. This corresponds to corks whose TCA quantity is less than 0.5 ng/l (Amorim Cork, 2020b). Therefore, G1 stoppers are the only ones to be added to a sales order.

3.2 Quality control

The entire process of analysis described before is completely automated. From the feeding system of the modules, to the transportation of the stoppers to the correct place after classification, going through their analysis and classification, everything is human intervention free. Workers only take care of providing the correct stopper lots to the feeding system and moving the classified corks to further operations in the factory, such as washing and packaging. Aside from this, the human role in the process is resumed to controlling and supervising the modules, adjusting any necessary parameters of the system to have it properly functioning.

3.2.1 Control indicators

Being an automated process, does not prevent completely the system from presenting its issues and failures or from becoming defective over time, potentially resulting in incorrect classifications. These issues subdivide in three different categories:

- Electro-mechanical failures, which essentially correspond to problems related to the physical components of the unit. The main examples are damaged seals, valves, injectors, columns or insulating components, cork debris inside the chambers or columns, or degradation of the stationary phase inside the columns;
- Software failures, that either translate into a software's inability to correctly calculate the TCA concentration in samples (the signal presents a wide variety of curves and shapes) or into unrecorded analyses due to insufficient computational power;
- Human errors, which occur when an employee does not set the system's parameters (pressures, temperatures, flow rates, etc.) appropriately.

To ensure the machines that the company are functioning as expected, there are some specific metrics that can be used, the Key Performance Indicators (KPI). Of all, the most relevant at NDtech is the Overall Equipment Efficiency (OEE). First introduced by (cite Nakajima 1988), it is calculated as:

$$OEE = Availability * Performance * Reliability$$

This indicator measures how well an equipment is working when compared to its full potential. The $Availability$ factor accounts for how much the equipment is being utilized considering how available it is for use. The $Performance$ factor accounts for how much time the equipment takes to perform a full production cycle, compared to the least amount of time possible it requires to do it. The third factor, $Reliability$, measures how well the actions are being performed by the machines,

i.e., how reliable are the machines performing the required tasks. In this work, *Reliability* is the factor of interest, corresponding to the percentage of corks that the machines correctly classified, from the totality of corks the machine analysed.

Reliability at NDtech is currently around 99,68% for the generality of the lines/machines, which means the machines are highly reliable, but there is still space for improvement. It is calculated as:

$$Reliability = \frac{TP + TN}{N}$$

Where, TP, the number of true positives, is the amount of corks correctly classified as not G1. TN, the number of true negatives, is the number of corks correctly classified as G1 and N the total number of corks the machine analysed. Besides TP and TN corks that have been classified can also be false negatives (FN) or false positives (FP). False negatives correspond to stoppers whose TCA levels were undermeasured. This means that, a cork that should have been classified as G2 or G3 was classified as G1, i.e. it was falsely assumed that it did not present detectable amounts of TCA. False positives correspond to the opposite case, when a cork that should have been classified as G1 is classified as G2 or G3, i.e., it was falsely assumed that it presented amounts of TCA that are too high. In both cases the company is impaired. With false negatives Amorim may be selling contaminated corks to its clients. This would be a serious scenario that can hurt the relationships with clients and potentially resulting in monetary penalties in favour of the client, because NDtech's business is based on the guarantee that every stopper they sell has an undetectable quantity of TCA, if no quantity at all. With false positives, the company is missing on the opportunity to sell perfectly acceptable stoppers, thus losing money as well.

Therefore, by reducing FP and FN, we assure that TP and TN rise, thus enhancing the Reliability metric as well. In other words, the number of false negatives and false positives must drop for the Reliability to rise. However, because of the specific design of the process at NDtech, the calculation of the Reliability must be adapted. First, because all modules in a line share the same stopper transportation system, there is only one basket for each class per line, instead of existing one for each module. As such, in each line, each of the baskets receives stoppers from all modules. Therefore, when a cork is extracted from one basket to assess whether it was correctly classified or not, it is impossible to know which of the line's unit analysed it. This way, the Reliability is calculated per machine, and not per module. Secondly, because only G1 stoppers are sold to the clients, the company does not have the habit to verify its G2 or G3 stoppers for quality control purposes. Therefore, at present, there is no data regarding False Positives or True Positives. It is simply assumed that all corks classified as G2 or G3 were correctly done so, and thus, all of them correspond to True Positives, not existing False Positives. Lastly, because the units necessarily classify most of the stoppers correctly, it is easier to calculate the Reliability using the number of FN, which is a much lower number than TP or TN. With this in mind, the final expression for the Reliability of a line corresponds to:

$$Reliability = \frac{TP+TN}{N} \Leftrightarrow \tag{3.1}$$

$$\Leftrightarrow R = 1 - \frac{FP+FN}{N} \Leftrightarrow \tag{3.2}$$

$$\Leftrightarrow R = 1 - \frac{0+FN}{N} \Leftrightarrow \tag{3.3}$$

$$\Leftrightarrow R = 1 - \frac{FN}{N} \tag{3.4}$$

$$\tag{3.5}$$

Considering all this, it is important for the company to control the system, not only to prevent the occurrence of failures, but also to detect possible errors already occurring in the measurements and improve the machines' reliability. Amorim has then come up with a control process which divides in three parts: pre-analysis control, control of the process during the analysis and post-classification control. Note that this does not include programmed maintenance inspections to the equipment nor post-failure maintenance interventions, that naturally must be performed to get the equipment back in function. These controls have been fundamental, playing a fundamental role in the inspection and supervision of the process, mitigating the appearances of FN and FP.

3.2.2 Pre-analysis control

This control is performed when a new lot of stoppers arrives to NDtech. A sample of the lot is removed for individual analysis to make sure that the lot corresponds to the expected specifications. Lots arriving to NDtech are expected to have a low percentage of TCA contaminated stoppers and thus if the analysis of the sample contradicts this assumption, the lot is rejected and will not be analysed at NDtech. This ensures that when a lot goes through NDtech, its acceptance level (defined as the percentage of G1 stoppers) is high – at least 80%. If this does not hold true when analysing the stoppers in the chromatographic units, it is assumed there is a problem with the equipment because the lot has already been validated.

Though, at the time of writing, this control is not performed anymore, several records of the analysed database still have gone through it. The reason why this step was discontinued is because currently, every lot coming to NDtech from other production units of Amorim is assumed to have passed special treatments - called VSR - that ensure the lot has the acceptable specifications. Thus, it becomes unnecessary to validate the lot again.

3.2.3 Control during analysis

Also referred to as equipment control, this second control is performed in the GC units, checking (and correcting) potential sensor errors. In every 12-hour shift, operators perform this control over every module four times, with two to three hours apart between each time. Basically, workers

in the factory inject a TCA solution in the chambers of the modules. By doing so, the unit measures the concentration of TCA of the injected solution. They then compare this measurement with what they know was in the solution. If the module's measurement does not match the expected values, the worker can adjust the process' parameters, recalibrating the module. This is repeated until they make sure the equipment is making correct readings.

However, it is important to understand that the four controls performed in every shift are not all equal. The first one, 8G3 INT control, consists in introducing a solution with a TCA concentration of 100 μg/l (15μl in each chamber), in four alternating chambers (odd or even), and then repeat it using the chambers where the TCA was not injected the first time. The second and fourth controls are the 8G2 controls. In this case, the solution has a concentration of TCA of 2 μg/l, and is injected in the eight chambers sequentially (30μl in each chamber). The third control is the 4G2 control and it is similar to the 8G2 controls, but this time only four consecutive chambers are tested, allowing the evaluation of the analytical columns (each shared by two chambers).

This way, G3 controls are essentially used to discover problems of carry-over - this concept is addressed below, but it corresponds to cases when the sample from one chamber influences the readings of another chamber's sample. In these controls the area read must be greater than (or equal) to 100 mV.s. The technicians also compare the peak times of all TCA readings (from all chambers) of a single module. If one of these peaks is not aligned, it can be inferred that a given chamber potentially has a problem. Alternatively, if there is a column problem, the peak appears offset in the two chambers attached to it. On the other hand, G2 controls, aim to detect sensitivity flaws - here the calculated area is evaluated and must be within the limits (15 mV.s and 30 mV.s).

This is a very important control for the company that allows the firm to find errors related to their units' sensors, which may influence measurements of the TCA concentration levels of the corks. This is sure to reduce the possibility of occurrence of both FP and FN. Because of its importance, this type of controls is performed regularly (every two to three hours) and without stopping the production runs. Thus, contrary to the other two control procedures, this one directly affects the data available for analysis. The system registers the measurements performed during the controls without distinguishing it from a regular cork in analysis. Therefore, it is important to acknowledge that part of the data corresponds to controls and not to regular analyses.

3.2.4 Post-categorization control

The last control checkpoint for the corks is done after these have been classified. Because Amorim only commercializes G1 corks, they perform a tight control over this group of stoppers to ensure that they only sell what they advertise.

At the end of each shift, each machine (set of six modules) will have classified a certain amount of corks as G1. From that group of corks, a random sample is taken for analysis. In more concrete terms, for every 12-hour shift, 10 corks (20 per day) are randomly selected for inspection from each machine's G1 classified stoppers -10 stoppers per shift per machine. This inspection serves to measure the quantity of TCA in those selected corks using a different process from the one at NDtech. The company uses this other method because they consider it more accurate and

effective. However, it takes 24 hours to analyse one cork and thus it is not suited for mass analysis as it is required at NDtech.

With this, the organization aims to verify the percentage of false negatives in the sample. If at least 1 of the corks (10% of the sample) turns out to have an unacceptable TCA level the entire production of the machine for that shift is rejected under the assumption that too many corks were incorrectly classified by that machine. In other words, the sample presented 10% of FN, which means the Reliability was 90% during that shift, considering FP do not exist. With such a "low" value, it is clear why the organization decides to reject the production.

3.3 Available dataset

With such a complex and automated process, data extraction from the system is a precious resource for analysing the entire operation. Thus, the company has created a database that registers several parameters of each of the readings of the modules of NDtech. Some of the most important features presented in the data are the identifications of the Line, Module and Chamber where each cork was analysed; the Lot from where the cork was taken; the TCA concentration and peak time of each analysis. A more detailed description of the dataset is provided in the next chapter.

Nonetheless, some important characteristics of the dataset must be addressed at this point because they contribute to the problem, namely the quality of the data available for use which is very poor. In the first place, some attributes are being manually inputted to the system with no standard form of registering this information. Each technician registers the data in its own way, leading to the appearance of many values that are different but correspond to the same concept, i.e., inconsistent data. Furthermore, the worker may only update such parameters a while after they are changed, or even completely forget to do so. This causes some records to have incorrect values in some attributes, which may be quite difficult to correct.

Secondly, some important features that could be useful to analyse are missing, as for example, sensor's parameters such as gas pressure and flow rate, torque of the motors controlling the valves, temperature in the chambers, columns, and detectors. External environmental factors, as temperature and humidity are other variables that are not included in the dataset and may influence the classification of the stoppers. Another defect of the dataset owes to the fact that it includes incorrect data due to software errors. Missing values in some observations' attributes, duplicated records and impossible values are some of the issues reflected in the database that arise from software errors. Another example of this is when there are missing records, i.e., when an analysis was performed but its information was not added to the database.

Lastly, data includes records registered during the controls, which is another source of contamination of the data. When the equipment control is being performed, the unit keeps adding information to the database, not indicating that the records do not correspond to analysed stoppers, but actually correspond to the injection of control TCA solutions.

3.4 Known flaws of the system

Naturally, over time, the company has developed a sense to identify the most common problems that arise from the failure issues already mentioned (electro-mechanical, software, computational, etc.) The analysis of the chromatograms created by each module's reading, and the comparison between values and trends measured within the same cycle and between different cycles has contributed to the know-how of the company and of the operators, which can now diagnose the most common flaws regarding the machine's performance. These are: Chamber Flow, Carry-Over (Chamber or Cyclic), and Peak Max Time.

1. *Chamber Flow*: this issue refers to the cases when the flow rate of the carrier gas or the mobile phase through its route inside the module is too low. This may happen due to improperly sealed chambers, leakage (in valves and columns) or debris clogging the path of the analyte on its way to the detector. In data this is manifested when a number of consecutive TCA readings is low for one or more chambers of the module, when compared to the average of the TCA values of other corks from the same lot.

2. *Carry over*: occurs when gas coming from one chamber remains in the module for too long, affecting the measurements of subsequent readings. There are two types of Carry-Over.

   (a) *Chamber CO*: this corresponds to the case when TCA coming from a chamber is not purged from the conduits inside the module after the cork in that chamber has been read. Instead it remains inside, accumulating with the TCA coming from other chambers that share the same routes to the detector (one chamber contaminates another). This results in a higher amount of TCA read for a sample than there truly exists.

   (b) *Cyclic CO*: in the case of Cyclic CO, instead of one chamber affecting others, it affects itself over two or more cycles. It may have the same causes as Chamber Carry-Over, but in distinct parts of the course of the gas. In essence, the TCA of a particular analysis is maintained in the path and accumulates with the next sample in the same chamber, resulting in higher readings of TCA concentration than reality.

3. *Peak Max Time*: the TCA peak time may vary from reading to reading due to changes in the system's several parameters such as the flow rate or temperature, or even due to the variation of TCA concentration between stoppers. This problem arises under the form of an unexpected displacement of the TCA curve, resulting in an integration within limits that do not correspond to those of the TCA curve. This will ultimately lead to an incorrect measurement of the TCA concentration in the sample.

Carry-over errors are directly related to the appearance of false positives causing high costs due to the reduction of the production yield (whole lots may be scrapped). In contrast, the chamber flow can result in the appearance of false negatives which also generates financial losses (including reimbursements to customers) and lost reputation (due to the perceived lack of product quality).

Finally, the peak-time fault issues indicate possible sensitivity flaws or the need to recalibrate the equipment. This fault may lead to both false negatives and false positives.

3.5 Currently implemented measures

Using the data previously described, and the experience of the team leading the operations at NDtech, some basic rules have been created to detect the four primary issues mentioned above. In other words, a set of four rules (one for each of the four main issues) analyses the data extracted from NDtech' s modules, identifying whether each record corresponds to a case where the measurement was affected by one of the issues, or if it is a normal reading. Currently the results of these classification rules are being displayed to technicians in a near-real-time context. This way, each technician evaluates that information and can take action over the units if necessary in his/her opinion.

However, these rules are measured and applied in the simplest forms. Several failures may happen simultaneously, resulting in complex patterns. Besides, a variety of other problems are not contemplated by the rules. For example, unexpected peak times or the non-existence of a peak are not examined by the rules. Instead, these issues are visually diagnosed by the operator in the floor, using the chromatography curve graphs produced by the software. Naturally, this is inefficient because operators cannot possibly check every graph from every measurement in every module, leaving many without inspection. Also, the rules are not sensible to the aforementioned issues data quality. If there are any missing analysis, for example, the rules are oblivious to it, carrying its calculations as if there was no problem.

In addition, the design of the four rules is based on the company's knowledge and therefore, follows several strong assumptions that are not statistically verified. This way, it is difficult to define the parameters and thresholds that support their implementation. Furthermore, the dynamism of the units' analysis conditions (degradation of materials, several different stoppers are measured, etc.) directly contrasts with the static nature of the rules. The lack of flexibility resulting from static parameters leads to low performance of the rules, resulting in a great number of false alarms, which can only be verified manually through human inspection.

Also, as it will be shown ahead in this document, every machine has a different behaviour, even when compared in the same conditions. These differences are even more accentuated when the comparison is performed between modules. This way, applying the same rules, with the same parameters to the data coming from every module is not accurate or effective and leads to poor results. On the other hand, with such a wide variety in behaviours and conditions, the task of fine-tuning the rules is very challenging, especially if it needs to be made individually for every single unit.

3.6 Project proposal

The business model of NDtech forces the organization to guarantee the absence of stoppers contaminated with TCA in their sales orders, while maintaining high production rates to respond to the existent demand. For this, it is necessary to guarantee that the GC-ECD units are functioning properly, or, in other words, that they keep a high OEE. This implies keeping high reliability levels as well, which in turn is assured by a low number of false negatives and false positives among the stoppers classified by the modules.

As detailed along this chapter there are many issues that restrain the company from achieving higher reliability levels (and low percentage of FN and FP). In the first place, the low quality of data greatly undermines analysis conducted over it. It is necessary to have a cleaner, more correctly labelled dataset that allows for better interpretation of the irregularities that bias the classification of the stoppers. To achieve this, it is necessary to normalize data entry, preferably automating that procedure and limiting the manual entry options.

The current data analysis method is inefficient. Besides not being sensitive to data quality issues, it is not adaptable to the machine's conditions. For example, the rigid thresholds and parameters of the defined rules are not flexible, which compromises the quality of the analysis in a set of units that behave so distinctively, as it happens in NDtech. This leads to the misclassification of a lot of records, whether as normal or anomalous data. Therefore, it is crucial to define an analysis approach that is both robust to eventual errors and issues in data as well as adaptable to the different contexts and behaviours of the machines.

Together, issues in the dataset and in the analysis method contribute to erratic information being provided to the technicians, which are ultimately responsible for adjusting and calibrating the modules in order to guarantee its correct functioning.

With this in mind, some nuclear requirements become clear:

- Creation of standard and automated procedures of registering information in the dataset, reducing the entry options for the technicians.;
- Creation of a data preprocessing framework, that tackles data quality issues and that can serve as a starting point for preprocessing actions in future studies of NDtech's database;
- Ensuring the application of a data analysis method that identifies anomalous patterns in the data collected from the equipments to provide useful information to the companies experts and that can potentially support the currently implemented rules or replace them at all. This approach should be both robust to the dynamics and variability of the context and behaviours of the GC units, as well as sensible to data pollution issues.

In order to achieve these goals, the project here portrayed is committed to analyse the data produced by NDtech's modules in order to give response to the above-mentioned necessities. A global analysis in the first place will enable the identification of the problems that the dataset has regarding its quality. This will then allow for the suggestion of some actions that may help

reduce the causes of such quality issues. Based on those suggestions, the standard procedure for registration of data in the database can be formulated.

Also, using data mining techniques to preprocess the data will help surpassing the data quality problems, allowing for a subsequent data analysis to be based on realistic high-quality data. The set of steps of the preprocessing procedure can then be used to create a framework that can serve as starting point for preprocessing in any future analyses to the data.

Finally, the analysis of data itself will extract patterns, providing information regarding the data observations and classifying them as normal or anomalous. The analysis process is expected to achieve results that demonstrate if the used technique is capable of supporting the currently implemented rules or replacing them at all as the method to detect incorrect classification of the corks. A detailed observation of the identified patterns will also allow the identification of the most common failures of the equipment (whether known or yet unknown), allowing for the categorization of such patterns.

These actions are sure to enable a more efficient support of the periodic manual equipment controls in a first phase, and later, help replacing them with spontaneous controls. This will improve the calibration and measurements of the process, paving the way for a more thorough monitoring of the system. As a result, the reliability of the machines will be improved (even more), bringing financial benefits for the firm.

Chapter 4

Methodology

This chapter presents a detailed description of every step performed in the analysis of the dataset. First, the main software programs used to perform the analyses are introduced. Then, the data set is described and summarized, including an overall presentation of the data structure. After this, the preprocessing stage is detailed, going through each step of that procedure. Additionally, the study of relations between variables and their relative influence in the classification of the stoppers is also reported. Next, the procedure to select the best sample of data over which to perform the detection of anomalous patterns is presented. Lastly, the techniques chosen to detect patterns in data are presented, including the procedures to adapt data to the technique.

4.1 Utilized software

Before starting to work the data it is needed to define which tools will be used for such purpose. In other words, what is the software that is going to be used to perform the several steps of the analysis.

Amorim stores the data using Microsoft SQL Server Management Studio (MSQLS), and thus, the data was supplied using that database. This software is suitable for accessing data, which is done through queries in SQL, but it is not indicated for data analysis. For that reason, it was opted to work data using Python, which is a free programming language, with an open development community that provides many packages that are ideal for the type of work needed in this project.

However, because of its continuous development, this software is frequently updated, resulting in the existence of many versions. In addition, when a new version is released, not all packages are ready to be used in the new version, which poses a dilemma. A newer version means that the software has less bugs, is more powerful, faster, and up to date. On the other hand, many important packages may not be available to use in the newer versions because these are individually updated at different paces. For this reason, it was opted to use Python's version 3.8, which allows the use of most relevant packages and is still a very recent version (the most recent is 3.9).

Another aspect to consider, has to due with the large quantity of data and restrictions in the computational power available. In these conditions Python is not very efficient at transforming the

dataset, namely during preprocessing tasks. This way, Python and MSQLS were jointly used in the project. The first, was used to perform the actual analysis of data, such as analysing variables or detecting patterns. The latter was used to perform data transformation tasks and obtaining data samples when needed.

4.2 Dataset description

Before starting to analyse data, preprocesing it and find eventual patterns, it is important to first understand what type of information and structure the database presents. The main database represented roughly two and a half months of production, from early September 2020 to late November 2020. It contained 21,058,684 records, evaluated using 32 different variables, amounting to around 4.5GB of data, and constituting, therefore, a Multivariate Time Series. Each observation corresponds to one chromatographic analysis, that is, a unique analysis to the TCA concentration on a single cork stopper. Thus, each variable either corresponds to a characteristic of the analysed stopper, to one condition of the analysis or to a result of the analysis.

In a primary observation and with the help of the company's experts, it was possible to note that many of the 32 attributes were irrelevant or redundant for this project. In other words, these features either expressed information already provided by other variables or did not present usable information at all. For that reason, only 15 variables were used to advance further in the analysis processes.

Variable	Description
AnalTime	date and time of the analysis
PunitLine	unit that performed the analysis
Chamber	chamber where stopper was inserted
Line	line wher the unit was placed
Punit	line wher the unit was placed
Shift	identification of the daily shift
Mode	configuration of the unit
State	configuration of the unit
Lot	lot from where the stopper was taken
Class	aesthetic class of the lot
Calibre	dimensions of the stoppers of the lot
PeakValue	detected TCA concentration
PeakTime	time of the occurence of the peak
DeltaTime	duration of the analysis
Quality	final classifcation of the stopper

Table 4.1: Brief description of the relevant features

As observable in Table 4.1, these 15 features can be divided in three groups. The first is composed of eight features that provide information about the conditions of the analysis, i.e., where, when, and how was the observation recorded. The first of these is the only non-discrete categorical

variable of the group, being instead, a continuous attribute. It is the 'AnalTime' variable, and it indicates the date and time of the analysis. The 'PUnit' and the 'Chamber' features identify in which unit and chamber was the cork placed for analysis, respectively. Next, there are two variables that indicate in which line was the module placed when it analysed the stopper referenced in each observation. They are the 'PunitLine' and 'Line' variables and though they present exactly the same information, they are not equal. In other words, their values do not match for every record (this issue will be addressed further ahead in the document). This variable is important for the company because sometimes they change the line of the units, that is, switch units between lines, and this allows to keep track of those permutations. Another variable in this group is the 'Shift'. This binary variable indicates in which of the two daily shifts the analysis was performed. Lastly, the 'Mode' and the 'State' attributes indicate how the unit was configured when it produced the record. For example, if the machine was being stopped, initiated or in full production mode.

The second set of features concerns the variables that provide information about the intrinsic characteristics of the corks. Three categorical attributes make up this group and they are the 'Lot', the 'Class', and the 'Calibre'. The Lot variable identifies the lot from where the stopper was retrieved; the Class variable identifies the aesthetic class of the stopper; and the Calibre feature indicates the dimensions of the stopper (height and diameter of the cylinder).

The last four variables are the actual results of the chromatographic analysis. The 'PeakValue' attribute indicates the value of the peak generated on the detector by the sample, used to compare against the thresholds and classify the stoppers. The 'PeakTime' attribute indicates the time of appearance of the TCA peak. As mentioned before, after the separation of the compounds in the analytical columns, the detector registers the variations in the signal provoked by the arrivals of each of the compounds. This variable thus presents the time (in seconds) where the maximum density of TCA flowed through the detector. The 'DeltaTime' variable indicates how long did the analysis of each sample took. Previously, it was mentioned that with the modules' design each analysis took 15 seconds and thus, values of this attribute are expected to be around this value. The fourth variable in this last group, contrary the other three, is not a continuous numerical attribute, but actually categorical. It is the 'Quality' and is the ultimate output of the analysis, indicating to which of three categories was the stopper assigned to, G1, G2 or G3, based on the value of the 'PeakValue' attribute and its comparison against the defined thresholds.

4.3 Preprocessing

Having clarified what is the type of information carried by the relevant variables of the dataset, the actual manipulation and inference process of the data can begin. As reported in the second chapter, the first stage of a project of this kind is the preprocessing stage, which has several steps. In this subsection, the main actions that were performed under the preprocessing steps are described: data summarization, data reduction, data cleaning and data transformation.

4.3.1 Data Reduction

Contrary to the usual sequence of steps of the preprocessing phase of a data analysis project, in this case, data reduction was performed prior to the summarization of the data. This is justified by the fact that, as seen above, the initial understanding of the dataset already served to conclude that some features were not necessary. The data reduction procedure simply consisted of eliminating those variables, which as mentioned in the previous section, were established as not being of interest. In other words, the dataset was reduced, leaving only the 15 relevant variables to work with. Because it was such a simple execution it will not even be mentioned in the next chapter, which presents the results.

4.3.2 Data summarization

Serving as a baseline to the definition of the subsequent preprocessing actions, data summarization is useful to assess data quality issues. In that respect, studying each of the dataset's features allows to check the level of impurity of each one. Furthermore, it serves to verify whether the values of the different attributes conform to the expected values and representation forms. Considering that, an individual statistical study of each attribute was performed. This way, using simple statistical metrics and visualization tools, several key results were obtained, crucial to the next tasks of the preprocessing phase, namely data cleaning.

As the main conclusions, it was verified that most of the categorical attributes presented serious quality issues that required thorough analysis and cleaning. Impossible values, inconsistencies, and noisy values populated the database with many undesirable impurities that needed removal or correction. All these problems are currently blocking the implemented rules, contributing to their poor performance in detecting anomalies.

4.3.3 Data Cleaning

The cleaning step mainly consisted of the correction and elimination of the records previously determined as having one or more attributes incorrectly written during the data summarization stage. Once more, the approach was performed one attribute at a time and thus the description of the actions in this stage will also follow that logic.

Mainly, the intervened attributes were the categorical features, eliminating and correcting the impure observations. Additionally, the issue of the redundancy of the variables 'PunitLine' and 'Line' was addressed at this stage. The variables were compared against each other and against the real values they should present, allowing for the elimination of the most damaged feature and the correction of the other.

4.3.4 Data transformation

Besides discarding or correcting the initial features, some new variables were also created. It was identified the potential usefulness of some characteristics of the records, and thus, new

features were created. These served the purpose of either helping the people of the company better understand some characteristics and trends in their modules' analysis, or to help discriminating eventual patterns that can be found in the dataset. This way, only the creation of new attributes was performed from the list of actions encompassed by the data transformation step reported in Chapter 2. However, the transformation of data is not finished here, because, as detailed ahead, other steps of the analysis required for it.

The created variables provided several new informations that are relevant to the analysis. Namely, the origin of the lot, i.e., the company from where it had arrived to NDtech was now reported in a feature constructed using information of production orders documents. Also, the information about whether the lot was VSR treated or not was now presented as well. Lastly, an error variable showed the most commonly found issues of the available dataset regarding values that would be unexpected in normal conditions, but can appear in certain conditions. In other words, these issues may be indicative of failures in the units physical components or software.

One last detail about this step was that a change was made in the naming of some variables. Specifically, 'AnalTime' was changed to 'AnalysisDate', 'PUnit' was changed to 'Module' and 'DeltaTime' was replaced by 'AnalysisDuration'. Although this change in naming was not necessary, it was performed simply to make more clear what the variable reported just by looking at its name.

4.4 Exploration of the clean variables

With the dataset cleaned and adjusted to advance to further investigations, studying the patterns of the units, a new analysis of the dataset was performed looking for hints on what would be the best next steps. Because of time and computational constrains, analysing such an immense dataset from the perspective of every feature would not be possible. Therefore, it was important to understand what would be the best sample to retrieve from the data that allowed for the most relevant conclusions to be taken regarding the patterns there found. This investigation mainly consisted on understanding relationships between variables and which of them were more useful to use for pattern analysis as well as which observations would be part of a sample capable of representing the variability of the original dataset.

4.4.1 Relationships between variables

This task consisted of the application of different statistical techniques to obtain information regarding the way variables interact with each other, providing useful information about their relative importance and influence in the results of the chromatographic analyses. Though some relationships were found between some pairs of features, there was no useful conclusion for the next steps of the analysis. This way, this step simply served to annotate those relationships between attributes.

4.4.2 Variable influence over the classification of stoppers

With the information gathered until this point, it was decided to investigate how the different variables influenced the classification of corks. Being the equipment where the detection of TCA occurs, studying how the results vary with the modules was important. Also, the influence of the line and of the chamber of the module could also contain relevant information, considering that each line is equivalent to a group of modules, and the chambers are intimately connected to the the functioning of the module. Also, according to the company it would only be natural that each lot would be somewhat different from the others, depending on its origin and on being a VSR lot or not. This way, it was also important to study the influence of the lots, origins and VSR characteristic over the results of the chromatographic analysis.

1. **Modules:**

   Starting with the modules, the first thing was to observe how different were the results of the modules analysis. In other words, how the units present different percentages of each type of cork. This way, it was studied the variation of each of the three 'Quality' classes for each module, to understand their influence on the probability of a stopper being classified as good for sale, that is, as G1.

   Next, to support the results obtained in the previous step, the same was done, but using the 'PeakValue' attribute instead of 'Quality'. This way, the mean 'PeakValue' was obtained for each module, allowing for another comparison between the different modules, in terms of their classification behaviours.

   Lastly, it was observed the percentage of annotated errors in the new variable that was constructed for each module. This would give an idea of which modules were more prone to present such errors, once more allowing for a comparison between them and possibly establishing a connection between the presence of such errors and influence on the classification of the stoppers.

2. **Lines:**

   Because the modules are grouped into lines, it was also interesting to perform the same analysis done for the modules, but from the perspective of the lines. This way, it could be inferred how the current distribution of modules over the lines is affecting the performance of the lines as a whole.

3. **Lot:**

   A very similar analysis to the modules influence on the classification of stoppers was performed for the lots. As the other big factor of influence of the likelihood of a stopper being classified as G1 or not, it was also important to verify the degree of such influence and compare it to the influence of the modules. For this reason, the variations of the values of the 'Quality' attributes and the variations of the average 'PeakValue' per lot was also investigated.

4. **Origin:**

   The same reasoning behind the investigation on the 'Line' attribute was performed for the 'Origin' attribute created earlier. Being the connector between different lots by identifying if they have the same origin or not, study how the different groups of lots could influence the classification results was also of interest. Therefore, the percentage of G1 stoppers obtained for each origin was observed as well.

5. **Chambers:**

   As a request of the firm, the average 'PeakValue' per chamber was also obtained. This served to compare the eight chambers and analyse if there was any of them with a greater tendency to have generally lower or higher peak values. This way, the design of the modules could be evaluated regarding its influence on the obtained analyses.

6. **VSR:**

   As another request of the company, it was observed how the fact of a lot being VSR or not influenced the classification results. Thus, once more, the percentage of G1 stoppers was obtained for both types of lots, VSR and non-VSR, allowing for the observation of the influence of this characteristic.

4.4.3 Selection of the sample

With these analyses performed so far, it got clear that the two biggest factors of influence of the classification of a stopper are indeed the lot from where the stopper is retrieved and the unit where it is analysed. This way, the sample to be extracted to detect patterns would either be based on specific modules, or on specific lots. The fact that the lot played a big role on the probability of the stopper being accepted was already expected and recognized by the company. However, for the modules, it could be the case that specific modules had mostly analysed the best lots, while others had mostly analysed worse lots, thus leading to the differences encountered between the units.

To figure this out, it was investigated how different modules evaluated a sample of lots. With that, it was observed that for each selected lot, different units had completely diverging levels of acceptance, i.e., percentage of G1 stoppers. With that, it was concluded that the module is truly a great independent influence on the classification of the stoppers. Therefore, the sample to extract to perform pattern detection should be representative of the diversity of modules.

To do this, a ranking of the modules was constructed. This ranking aimed at ordering the units by their likelihood of classifying stoppers as G1. In other words, it was possible to observe which modules had more tendency to classify stoppers as G1 and which had the least tendency to do so. With that, it was possible to select the sample retrieving three groups of modules. One group from each end of the ranking and a third from the closest to the middle of ranking. In total, the sample contained the entire data for 10 different modules.

4.5 Detection of anomalous patterns

With the sample selected, it was then possible to proceed to the analysis of patterns in the data. The detection of anomalous patterns that was performed can be divided in three groups. First, using DBSCAN clustering algorithm, anomalous patterns in the 'PeakValue' variable were identified for each of the samples' modules. This was the main approach to the identification of the anomalous patterns. Second, using a simple approach based on histograms, anomalous patterns in the variations of the 'PeakValue' signal were identified for some modules of the sample. Third, using the same basic approach with histograms, some anomalous patterns were obtained in the 'Errors' feature, also for a few of the modules of the sample.

The reason why the main focus of this detection was based on clustering is because this technique not only allows for the separation between normal and anomalous data, but it is also able to distinguish between different types of anomalies. This allows for a deeper analysis to the most frequent anomalous patterns, identifying their meaning and cause in the factory floor. Also, by identifying what the patterns correspond to, it is possible to establish comparisons between the selected groups of modules and to perceive which are the most common anomalies across all groups.

It should also be noted that the detection of patterns in each of the three approaches was performed using sliding windows. This way, the data from each module was, in a first phase, divided into segments. This was because the modules have production breaks along the period portrayed in the data. A chromatographic analysis performed before a maintenance stop and another performed after could not be included in the same window because the conditions of the analysis had changed. Therefore, by first dividing the data into segments, each run was separated and then the windows were constructed, paying attention not to mix data from each segment.

4.5.1 Clustering approach

Because using all segments of a module to obtain patterns was proving to be too computationally expensive and time consuming, it was decided to use only 60% of the segments generated for each module. Also, in order to make this approach more robust there were four steps performed before the actual application of the DBSCAN algorithm, aside from the separation of the data into segments and then windows. First, the size of the windows needed to be defined, and it was done so, by the number of cycles of the machine. This way, the smallest possible window would contain one cycle, or eight analysis, one for each chamber. There were three different sizes experimented: 4, 8 and 16 cycles per frame, each with a leap of one cycle, i.e., being displaced from the previous frame by one cycle, meaning the other cycles overlapped with the next frame.

Secondly, statistical features were extracted for each window. For each chamber, eight features were extracted from the 'PeakValue' attribute:

- mean
- standard deviation
- kurtosis
- skewness

- mean difference between consecutive analysis
- standard deviation of the differences between consecutive analysis
- kurtosis of the differences between consecutive analysis
- skewness of the differences between consecutive analysis

With this, each point/window became a vector with 64 dimensions (eight features $\times$ eight chambers).

After the extraction of the features in each window there was the third step, which corresponded to the application of PCA to the features. This way, the features were translated into uncorrelated variables and normalized using the z-score normalization method. This balanced the features because Clustering techniques are based on the distance between points. Thus, if one of the features is defined in a bigger scale it would dominate the creation of clusters. With the application of PCA, this problem was bypassed.

Lastly, before the application of the clustering algorithm, its input parameters had to be defined. DBSCAN algorithm required two parameters: 'eps' and 'min_samples'. The first corresponded to the distance between two samples for one to be considered as in the neighbourhood of the other. In other words, if two points were separated by a distance greater than 'eps' they would be placed into different clusters. The second, is the minimum number of samples that a cluster must have to form a cluster.

Defining 'min_sample' was simple. It was considered that a single point could be an anomaly different from all others, and thus, a single point was necessary to create a cluster, i.e., a type of anomaly. 'min_sample' was thus defined as equal to 1.

On the other hand, 'eps' was a parameter harder to define since it varied from situation to situation. To obtain the best value for this parameter an iterative cost function was created. Naturally, anomalies occur with much less frequency than normal data points. This way, the biggest cluster would consist of the normal data points. Also, to allow for the identification of the most frequent and relevant patterns, it was required that the biggest classes of anomalies were composed of the highest number of examples possible. Thus, the cost function constructed found the 'eps' value that would maximize the number of points for the second biggest cluster while at least 80% of the points belonged to the biggest cluster. With this, it was guaranteed that 80% of the data was normal while the most relevant anomalous cluster would have a significant number of examples.

This cost function was used twice. The first time, the function incremented the eps value by 0.5 in each iteration, starting at this value and terminating with a maximum 'eps' of 30. This allowed to obtain the best 'eps' with an error of 0.5. In the second time, the function incremented the eps by 0.1 in each iteration, starting with an eps equal to the 'eps' found in the first run minus 0.4, and ending at the same 'eps' plus 0.4. This way, the best eps was found with a precision of one decimal place.

With this, the main anomalous patterns were found for each module and each group of modules. The comparison of results between each of them allowed for the drawing of interesting conclusions, bringing highly valuable information to the company. The approach, has however a

disadvantage. When a cycle in a frame has a missing value, the method can not handle it and thus, discards the whole cycle. This results in the potential loss of of the possibility to identify some patterns.

4.5.2 Secondary approaches

As mentioned, the two secondary approaches were based on the construction of histograms. Both share some of the initial steps of the clustering approach, namely the division of the data of the units into segments and the construction of the frames. However, in these approaches only 3 modules were used, one of each group of the selected sample of modules. In turn, all the segments were included, not just 60% of them.

4.5.2.1 PeakValue variation signal

For the first of these secondary approaches, it was used the 'PeakValue' attribute as in the clustering approach, but the features extracted were different. Indeed, for each chamber it was extracted the variation of the signal of the 'PeakValue'. In other words, each peak value was replaced by the difference of that peak value and the peak value of the previous analysis in the same chamber, and then by the signal of that difference. For example, if chamber one presented a higher peak value in a certain cycle than it did in the previous cycle, then the peak value would be replaced by he value '1'. Similarly, if the value was lower or equal to the one registered in the previous cycle then '-1' and '0' would be respectively registered instead. In this approach, it was also possible to keep cycles with missing chambers by noting the missing chamber with the value '2'.

In this approach it was used a frame size of two measurement cycles and a displacement of one cycle. This way, each frame would have 16 values, two per chamber. Then, by turning these values into strings, the dimension of the frame was reduced to 8, where each string contained two consecutive variations of one chamber. As an example, assume frame X was equivalent to an array of variations: [1,-1,1,-1,1,1,-1,1,-1,-1,0,-1,1,-1,-1,1]. Then it was transformed into an array of 8 strings: ['1-1', '-1-1', '10', '-1-1', '11', '1-1', '-1-1', '11'], each representing two consecutive variations of each chamber. This way, by working with strings it was not possible to use clustering algorithms, which are based on distances, thus requiring numerical inputs.

After obtaining the strings of all frames in all segments, it was constructed a histogram to observe the distribution of the strings. Note that there are limited possibilities that a string can have in this case: with only four possible values in each (-1;0;1;2) there were only 2^4 possibilities, that is, 16.

4.5.2.2 Error patterns

The main difference between this one and the previous secondary approach lies on the fact that instead of the PeakValue attribute, it was used the 'Error' variable that was created before. Also,

instead of using frames, each cycle was treated as a data point. In practical terms it is as if the frame size was equal to one cycle.

This way, the sequence of values of the ‘Error’ variable in one cycle was also converted to a string. In this approach, the problem of missing records was also solved by indicating a value of ‘9’ when a missing record was detected in a cycle, avoiding the loss of the entire cycle (recall that values for the ’Error’ feature were defined between ’0’ and ’7’, thus ‘9’ was available).

The histogram that was then constructed presented the frequency of each possibility for the sequence of errors in a cycle. In other words, the most common sequences of errors in one cycle were able to be observed.

Chapter 5

Results discussion

This chapter presents the results obtained through the application of the methodological steps described in the previous chapter. Summarily, there are four core results to present, derived from the application of the methods described before. There are results concerning the initial analysis of the data and its quality issues; results from the preprocessing steps performed over the raw data; results acquired from the investigation of the influence of the variables of the dataset over the final classification of the stoppers; and results obtained from the application of anomaly detection techniques in the clean dataset to identify anomalous patterns. The followed structure is in accordance with the order of the methodology steps described in Chapter 4.

5.1 Data preprocessing

As the first step for a project of this nature, the first results concern the application of the several steps of the preprocessing of the dataset. From the summarization of data to its transformation, passing through cleaning the main conclusions follow ahead.

5.1.1 Data summarization

Utilizing statistical methods to analyse the different features considered relevant in the dataset, the information obtained from each variable follows next. Annex A contains the graphical supports for this step.

- ***'AnalTime':*** simply representing the date and hour of the recording of each data point, this variable works as the identifier for the record as well. The only informations retrieved from this attribute at a global level regarding the dataset are the period of analysis, 08/09/2020 to 27/11/2020 and the longest period between consecutive analysis – 06h21min. Therefore, the database represents data collected from a period of approximately 81.5 days.

Variable	Oldest	Youngest	Max Difference
AnalTime	07/09/2020 23:59	27/11/2020 14:57	0 days 06:21:23

Table 5.1: Summary of variable 'AnalTime'

- ***'PUnit':*** indicating the unit where the analysis occurred, this variable was expected to present 60 different categories, each representing one of the 60 modules at NDtech. Observing the information in Table 5.2 and Figure A.1, the most important takeaways resultant from this initial analysis are:

  - Though 60 different categories were expected, data indicates that there are only 59 different classes for this variable, meaning there is one module that is completely missing from the dataset.
  - Unit 69 is the class/module with the biggest representation in the data, accounting for 3.3% of the records. However, this unit is not part of NDtech's lines, integrating, instead, another section of the organization. Therefore, it is wrongly included in the dataset and must be removed.
  - Unit 00 is another category that is present in the data but does not exist on NDtech. It is actually the value that the system assumes by default, not a real module. When a unit is interrupted (for maintenance purposes for example) and then rebooted, if the operator forgets to set this parameter correctly in the software before the module starts running again, the software assumes 'Unit 00' by default until it is corrected later. Therefore, the records in this class need to be analysed in order to allocate them back to the correct module, or eliminated if correcting is not possible.
  - With the presence of the labels 'Unit 69' and 'Unit 00', the number of modules actually missing rises to three.
  - Aside from classes 'Unit 69' and 'Unit 00', the histogram in Figure A.1 shows that there is a great difference in the representation of each module in the dataset. For example, module 21 has only 0.33% of records under its class, while module 17 is accountable for 2.11% of analysis, almost 6.5 times more. This corresponds to a difference of 373,465 records, and in a population of 57 units it is quite significant. This may result either from a real difference in the availability of the the modules, or from some modules not being always connected to the database.

- ***'Chamber':*** registering which chamber was used to inject the cork, this variable was expected to have 8 categories, one per chamber. However, in addition to the 8 chambers, there are two other classes in this variable, whose labels are '0' and '99'. Records belonging to these classes do not correspond to measured corks, but actually to software signals. When the module is initiated, it sends a signal with the value '0' for the 'Chamber' value. Contrarily, when it is interrupted, the signal carries the value '99'. This way, records in these

classes may be eliminated from the dataset because they do not actually correspond to a chromatographic analysis.

Also, it can be concluded from Figure A.2 that the eight actual chambers are represented almost equally. In other words, across all modules, the eight chambers perform roughly the same amount of analyses. However, odd numbered chambers are all slightly less represented than their even numbered counterparts. A possible explanation for some chambers being less represented may have to do with missing records. As explained, sometimes an analysis is performed but does not get registered in the database due to software errors. The reason why odd numbered chambers are the most affected may have something to do with the unit's design. Though there are 8 chambers, there are only 4 chromatographic columns, each shared by two chambers. Each chamber's pair is the one whose place is 4 chambers ahead. This way the chamber pairs are 1/5, 2/6, 3/7 and 4/8. This means that odd numbered chambers pair up with another odd numbered chamber. It may then be speculated that something in the design of the module leads to registration errors in the measurements where the separation of components occurs in the two columns connected to the odd numbered chambers. In other words, those two columns are more likely to result in failures.

- ***'PunitLine':*** as one of the variables that depict the machine/line where the module was placed when it performed each analysis, it is expected to present ten different categories, one for each line. However, it was observed that there are 11 classes in this variable, one more than expected. This owes to the presence of the class 'Line XX' which is the default value assumed by the system. Once again, this variable is manually defined, and if an operator forgets to set it, the system will assume the default value. Therefore, records under this class must be analysed in order to be properly reassigned to the correct 'PunitLine' class, or, if this is not possible, eliminated.

  Also, apparently some lines have a much greater use than others. For example, Line 7 is accountable for 2342537 (11.12%) analyses, while Line 4 is responsible for just 748279 (5.6%) analysis - a difference of 1594258, or5.52%. Again, this may be related to some modules not being connected to the database during the entire period of analysis, or having low availability. This can be observed in Table 5.2 and Figure A.3.

- ***'Line':*** as the second feature depicting the machine/line where the module was inserted when it performed a given chromatographic analysis, this variable suffers from the same problems as the first, 'PUnitLine'. Depicted in Figure A.4, there are two unexpected classes in this variable, '0' and '16' whose observations need to be verified in order to assess if they correspond to wrongly registered records, or if they should not even be part of the dataset. Also, as seen in the 'PUnitLine' variable, there is a considerable disparity in the representation of each line, with the difference between the top and bottom lines fixated at 1407751 (or 6.69%).

- ***'Shift':*** contrary to the observed features so far, the Shift attribute is quite clean and in conformity with the expectations (see Figure A.5). There are no unexpected categories and both daily shifts present roughly the same number records. Although there are slightly more records belonging to the first shift, this small difference is expected because it is on the second shift that programmed maintenance is performed in the units. When this happens, the units are stopped, not sending data to the database, and leading to the observed difference between the two shifts.

- ***'Mode' and 'State':*** these features serve the purpose of informing in which condition was the observation produced. In other words, it shows if the data point was created when during normal functioning of the units or if there was some error associated. In the first case, the Mode feature should present the value 'Production' and the State feature should present the value 'Running', which is what happens in 99.5% of the data. The remaining 0.5% indicate situations where the modules were not in their normal activities and thus should be eliminated (see Figures A.6 and A.7).

Variable	Nr. Classes	Modal Class	Mode Count	Min. Class	Min. Class Count
PUnit	59	Unit 69	694 766	Unit 00	53 513
Chamber	10	4	2 684 596	99	10 684
PunitLine	11	Line 7	2 342 537	Line XX	748 279
Line	12	9	2 701 467	16	62 893
Shift	2	Turno 1	10 850 528	Turno 2	10 208 156
Mode	8	Production	20 487 143	Low sensitivity-Punit	1
State	18	Running	20 497 667	FESTO	1

Table 5.2: Summary of variables depicting conditions of analysis

- ***'Lot':*** showing the lot from where the corks were retrieved, this variable is useful to make sense of data and confirm its veracity. By comparing the values of this variable with other documents of the company, namely Production Orders, we can check whether the values on the database correspond to reality, or not. As observable in Figure A.8, there was a great variety of lots used during the period of analysis, and a great variety on the representation of each of them as well. Furthermore, the majority class was '123456-A' with 5.76% of the records. As seen in previous categorical variables, this is the default value for this feature, not an actual lot. It must be corrected to reallocate the observations in this class to the correct lot. In addition, the 293 different lots (see Table 5.3 identified do not correspond to the real number of lots that were analysed at NDtech, according to other documents of the company. This is presumably due to lack of standardization in the introduction of this variable in the system, that leads to the existence of different values representing the same Lot. This matter should also be investigated.

- ***'Class':*** this variable shows the aesthetic class of the cork and can serve the same purpose of veracity checking as the 'Lot' feature. Besides, for each lot, every cork will have the same class, so it is useful to compare values between records to verify the existence of incorrectly

input values. Though the feature shows 56126 different classes (see Table 5.3), according to the NDtech's team, this value is much lower in reality, being possible to have 40 categories at the most. Also, there is an enormous difference in the appearance of each class in the database, with some classes having only 1 record.

It is then clear that, the fact of being manually introduced in the system leads to many different categories because each technician registers the Class as they please. Therefore, many of the classes observed are in fact the same. This variable needs to be verified in order to format its values to a normalized format. Figure A.9 presents the histogram of this variable.

- ***'Calibre':*** the feature Calibre follows the same logic as the Lot and Class variables, depicting intrinsic characteristics of the cork, in this case, its dimensions. As is the case of Class attribute, each Lot only has one Calibre of corks and thus, checking this variable's values is another way of verifying the correctness of the records.

  As Table 5.3 shows, there were 56204 different categories found in this variable, which is nowhere near the reality of the company. In fact, according to the company, there should be less than 10 different calibres found in the data. Once again, this disparity between reality and what data shows owes to this feature being manually inserted in the system and each technician registering a value on its own way, without a standard format. This variable must also be corrected to eliminate the variability in the representation of the same calibres.

Variable	Nr. Classes	Modal Class	Mode Count	Min. Class	Min. Class Count
Lot	293	123456-A	1 212 367	161616	1
Class	56 126	Extra	4 136 909	12.6600	1
Calibre	56 204	49x24	8 298 101	5.1733	1

Table 5.3: Summary of variables depicting characteristics of the stoppers

- ***'PeakValue':*** probably the second most important variable of the data set (behind feature 'Quality'), presenting the estimated concentration of TCA in the stoppers. It is by comparing this value with the defined thresholds that the cork is classified as G1, G2 or G3. The first thing that is noted on this variable is how unevenly distributed it is. The value for the third quarter is 4.80 mV.s, but the maximum value is 43,622.91 mV.s. Also, its mean and median are 7.32 mV.s and 2.44 mV.s respectively, but its standard deviation is fixated at 68.97 mV.s, almost 10 times the mean (see Table 5.4). As observable in Figures A.11 and A.12, data is highly skewed as well. In fact, it presents a skewness of 102.22, which means it is tremendously positively skewed, with a long tail to the right of the peak, i.e., very asymmetric.

- ***'PeakTime':*** denoting the time, in seconds, at which the peak of TCA was registered at the sensor, the values on this feature had forcibly to be lower than 15 seconds, because that is how long the entire detection takes. This holds true for the information shown by this

variable, with its maximum being 11.4 seconds, as shown in Table 5.4. Also, its mean is in accordance with the value expected by NDtech's team, around 4 sec. It is also interesting to note the considerable amount o records with a value of 0 seconds for this feature. This is an impossibility and corresponds to cases when there was a measurement error or a software error in registering this attribute. This is not a critical issue, however, because stoppers that had this issue are classified as G2, being stored for new analysis later. Figure A.13 shows the histogram of this variable.

- ***'DeltaTime':*** registering the duration of the detection (in seconds), this feature's values are supposed to always be 15 seconds. However, due to software issues and imprecision, values of 14 or 16 are also admitted as correct. Indeed, 15 seconds it the most common value for this variable, by far, encompassing 91% of the records. 16 and 14 seconds follow next, both with roughly 7% of the data. On the other side, around 4.3% of the data - equivalent to 879749 records - present a DeltaTime of 0 seconds. This is the result of software errors and in this case, corks are also classified as G2. Finally, the rest of the data, 88469 observations, present various values of 'DeltaTime', which also correspond to software errors. These situations are observable in Figure A.14 (the graph is presented as a bar chart instead of an histogram to facilitate the visualization).

  In more statistical terms, these issues have a significant impact, resulting in an average value of 23,554 seconds and a standard deviation of 9471209.8 seconds, which are values in complete disagreement with the objectives of the company. Note as well, that the highest DeltaTime recorded is 3,813,856,109 seconds, i.e., more than 44 thousand days, which is absolutely impossible. Naturally this variable is also considerably positively skewed – skewness is 402.48.

Variable	Mean	Std. Dev.	Median	Min.	Q1	Q3	Max.	Skewness	Kurtosis
PeakValue	7.317	68.966	2.443	0.0	0.23	4.795	43,622.910	102	34,762
PeakTime	3.733	1.281	3.6	0.0	3.1	4.4	11.4	0.063	1.79
DeltaTime	23,554	9,471,210	15.000	0.0	15.0	15.0	3,813,856,109	402	161,985

Table 5.4: Summary of the numerical variables that shows the resultant classification of the chromatographic analysis ('PeakValue' is measured in mV.s; 'PeakTime' and 'DeltaTime' are in seconds)

- ***'Quality':*** the most important feature of the dataset, presenting the final class to which the stoppers are assigned. Besides the three classes corresponding to the three final groups, G1, G2 and G3, other minor classes were found, that correspond to software registration errors, that should be eliminated (see Figure A.15). Class '0' is the major class, with 81.63% of the records and corresponds to G1 stoppers. Class '2' is the second most represented – 14.22% - and corresponds to G2 corks. Class '1' encompasses 4.15% of the observations and corresponds to G3 group. These values are in accordance with the company's estimations and expectations.

Variable	Nr. Classes	Modal Class	Mode Count	Min. Class	Min. Class Count
Quality	55	0	17 190 838	3	1

Table 5.5: Summary of the categorical variable that shows the results of the chromatographic analysis

With this description of the different features in the datasets the main weaknesses of the data became clear. Many of the categorical attributes suffer from an absurd variability in their values due to being manually inserted in the system. Also, the numerical variables are contaminated by inputs corresponding to software errors. These situations compromise the analysis of patterns in the data. Therefore, a thorough cleansing of the data must be performed, in order to obtain a final dataset that is representative of the reality of the chromatographic analysis performed at NDtech.

5.1.2 Data cleaning

Witht the summariztion of the data, it was possible to understand what were the factors causing the impurities in the data. This way, utilizing that knowledge, the data cleaning step of preprocessing phase was performed a variable at a time. Annex B presents tables that help visualize and understand the actions described along the process and their results.

- ***'PUnit':*** starting with this variable, the first issue to address was the absence of three units in the database. The reason for this was rapidly identified after consulting the team at NDtech. In fact, according to them, this absence was expected because of the three missing modules, two were in maintenance for the entire period of analysis and the third was never connected to the database. Therefore, it was natural that these are not represented in the dataset.

  Next, data labelled with 'Unit 00 or 'Unit 69' was investigated. As explained, these units do not exist at NDtech and thus the records contained in these groups had to be analysed. Using the Lot, Class and Calibre features it was tried to match these data points with data from other units, but it was not possible to establish with certainty that these data points belonged to any of the other modules. Therefore, it was opted to eliminate points belonging to Units '00' and '69'.

  Also, it was informed by the company that 'Unit 28' was functioning with an old and outdated software that was different from the software in the other units. Therefore, data produced by this unit was produced by slightly different mechanisms than data generated by the other modules. For this reason, data from 'Unit 28' was also eliminated from the dataset.

  With these actions, the number of classes in the database was reduced to 56, and the number of records to 20056633, representing a loss of 4.76% of the data.

- ***'PunitLine' and 'Line':*** regarding the line of the module, the first issue to be addressed was the choice between variables 'PUnitLine' and 'Line'. Since both carry the same information, one of them was redundant and therefore could be discarded. In order to make this choice,

the correspondence between modules and lines in each of the variables was studied and compared to the reality. In other words, the veracity of the information of each variable was verified. Table B.1 demonstrates that correspondence.

The table shows the number of records for each combination of Unit/PUnitLine/Line. For example, 399089 observations had the combination Unit 02/PUnitLine 1/Line 1 (written in blue). For this example, both features show that Unit 02 belongs to Line 1, which also corresponds to reality. However, this is not always the case, with the values of one or both features sometimes not being correct.

Also, it can be noticed that some modules appear in more than one line even when looking to just one of the variables. For example, 'Unit 62' has some records that belong to line 3 and some that belong to line 6 in the 'Line' variable. In theory this could happen, because the company does change the modules from a line to another. However, it was assured by NDtech's team that during the period of time captured in the data, no switches were made regarding the modules' positions. Therefore, each module must only have one associated line in the database.

Considering this, and assuming that the Unit variable was always correct, the error rate was obtained for each variable by calculating the percentage of records that had a label that did not correspond to reality. Note that the total number of records used for this calculation is not the same as the initially mentioned number of total records in the database (21,058,684). This owes to the previous elimination of the observations regarding 'Units 00', '69' and '28'. Because the error rate was lower for the 'Line' variable (9.59% versus 47.85%), this was the feature chosen to keep in the dataset, while 'PUnitLine' was discarded.

Lastly, with the best Line variable encountered, it was needed to correct the observations that led to the 9.59% error rate. In other words, for those records, the value of the 'Line' feature had to be changed to correspond to reality. Table 5.6 highlights the cases where variable 'Line' differed from reality and the actions that were taken to correct the dataset.

In short, each set of observations produced by a module and with an incorrect 'Line' label, was compared against data from the correct/real line using the variables 'Lot', 'Class', 'Calibre' and 'AnalTime'. For example, for the observations generated by Unit 01 but allocated to Line '0' it was obtained their respective lot, class and calibre and the period of generation of such records. Then, for the same period, the lot, class and calibre values were compared against those of records of Line 1 in the same period.

This approach led to the actions reported on the last column of Table 5.6. Observations of the first module with incorrect line values did not have any correspondence to the values in other modules from the first line in the same period and were thus eliminated. Data belonging to 'Unit 06' and Line 3 was the only exception to the rule that the annotation of the 'PUnit' variable was always correct. It was found that this data was in fact from line '8', but not from module 06, it was, instead, data from Unit 21. As for the rest of the entries of Table 5.6 it was verified that the attribute Line was the only thing incorrect and that those

Punit	Line Variable	Real Line	Action
Unit 01	0	1	eliminated
Unit 01	16	1	eliminated
Unit 06	3	8	changed to module 21 and line 8
Unit 10	5	9	changed to line 9
Unit 12	4	5	changed to line 5
Unit 25	7	9	changed to line 9
Unit 39	8	9	changed to line 9
Unit 43	3	10	changed to line 10
Unit 44	5	3	changed to line 3
Unit 53	9	2	changed to line 2
Unit 58	9	7	changed to line 7
Unit 62	3	6	changed to line 6

Table 5.6: Actions taken to clean variable 'Line'

observations had correspondence to data in the true lines. Thus, for these cases, variable 'Line' was simply changed to match reality.

- ***'Chamber':*** regarding this variable, the alterations to data were not many nor difficult. It simply consisted of the elimination of data points whose Chamber value was '0' or '99' because, as explained, these were not actual TCA measurements, but rather signals produced by the software to indicate start and stop stages of the modules.

- ***'Lot':*** as observed in the initial analysis, there were two main issues related to the 'Lot' feature. First there was the '123456-A' class that corresponded to the default 'Lot' value. Second, there was the question of the lack of standardization on this variable that resulted in the existence of 293 different lots in the dataset.

  Regarding class '123456-A' the approach was similar to that used to solve the previous issue of units allocated to incorrect lines (see section 5.4.2). This way, values from this class were divided using the respective unit, line, and time information. Then, data from the same line, but from other modules and generated in the same period was used to perform the correction. For example, a set of points from 'Unit 08' belonged to lot '123456-A' and was generated from 06/11/2020 to 07/11/2020. Then, because 'Unit 08' belongs to line '9' data from that period produced by other modules of that line was inspected. With this, it was possible to know the correct lot that was analysed by 'Unit 08' in those two days because it had to be the same lot that was analysed in other modules from the same line (recall that all units of one line always analyse the same Lots at the same time). This procedure was repeated until every data point from lot '123456-A' had been corrected.

  As for the issue of the lack of standardization, it was first identified which were the actual lots that went through NDtech at which time and in which Line using documents provided by the company such as Production Orders sheets. Then, with the sequence of Lots identified for each line it was possible to verify the dataset in order to assess if the information in

the database matched these other documents. Whenever it did not, the 'Lot' variable was corrected. However, this was quite challenging because of the wide variety of lots and lines, and because many times the information on the external documents was also flawed or not available with the required detail. This, together with the fact that these verifications and corrections had to be manually executed, resulted in a long duration for this process. Also, because of the inefficiency of the approach, in a few lots, data had to be eliminated due to lack of certainty on which was the correct lot, or corrected with some assumptions that may not be precise.

- ***'Class':*** the problem with this variable also resided in the fact that it is manually entered in the system in a non-standardized way, creating several categories that represent the same Class, that is, inconsistencies. To correct and rewrite this feature in a standardized way, it was investigated which were the values that this variable assumed.

  Previous data eliminations eased this task, lowering the initial number of 56,126 classes to 132. The classes were grouped and changed according to the correct standardized form of the actual class. The nomenclature used for writing the classes was 'GenC (variant) indication', with 'GenC' representing the general class or mixture of general classes, with every letter written in uppercase and with spaces between each part of the name of the class. For example, class 'S/1 (W 2) R15' had initially 10 different classes that corresponded to it. It is a class that corresponds to the mixture of corks of Classes 'Primeira' and 'Superior', with a '(W 2)' variant and 'R15' indication. With this procedure it was possible a reduction from 132 classes to 28 classes.

- ***'Calibre':*** variable 'Calibre' suffered from the exact same problem as variables 'Class' and 'Lot'. For that reason, the approach was also the same. In this case there was also a facilitation of the process caused by the previous data eliminations. In the initial situation, there were 56,204 different calibres found in the database, but when this issue started to be addressed there were already only 20.

  Contrary to the previous two features, this one has a fairly simply and standardized nomenclature. Corresponding to the dimensions of the stoppers, which are cylindrical, the regular calibre is written as 'hxd', where h is the height and d is the diameter of the cork. For this reason, most of the problem had to due with the 'x' being written in upper or lower case, and as it can be observed in table 5.7.

  Aside from that, only four of the erratic calibres required comparisons of its data and other data generated in the same lines for the same period. In the end, there were only eight different calibres in the database.

- ***'Mode' and 'State':*** As indicated, these variables indicate if the records were produced in normal conditions or not. Thus, the task regarding these features consisted of simply eliminating data that was not produced in regular conditions. With this, only one class remained for each of the features, which meant that these variables could then be eliminated.

Calibre	Variations
45x24	'45x24' * '45X24' * '495x24'
45x25	'45x25' * '45x25'
45x26	'45x26' * '45x26'
49x24	'49x24' * '49x24' * '49x24.5' * 'e/s(w1)r15'
49x25	'49x25' * '49x25'
49x26	'49x26' * '49x26'
54x24	'54x25' * '54x25'
52x24	'52x25' * '52x25' * '52/54x24'

Table 5.7: Variations of the true calibres in the data

5.1.3 Data tranformation

With the results obtained so far, and as mentioned in the previous chapter it was felt that the construction of new attributes could bring new and useful information. Therefore, the data transformation step performed at this stage merely consisted of the creation of three new variables: 'Origin', 'VSR' and 'Error'. (actually, three of the existing variables were also changed, but only their naming).

- ***Origin:*** the 'Origin' variable was created in order to associate each lot to its company of origin. Though the majority of lots analysed at NDtech came from within Amorim, that is, from other industrial units of the group, for some lots this was not the case. Namely, there were lots that did not belong to the company at all and were analysed at NDtech as services provision to other firms. There were also some lots that came from the laboratory, after being inspected, or were already going through NDtech analysis for the second or third time, being refined in each iteration. With this array of background origins, it would be important to study whether this factor of origin of the lot has any influence on the classification of the stoppers. The attribute was constructed using the information in documents provided by the company that related each lot with its origin.

- ***VSR:*** VSR was simply created to identify which records represented stoppers that suffered the homonym treatment. As indicated in chapter 3, the first quality control step involves sampling and testing the lots before they are analysed at NDtech. However, this has been discontinued due to the VSR technology, which should assure that the lots are already acceptable and do not need that initial verification. This way, the company also expected to have a higher acceptance level (higher resultant percentage of G1) for VSR lots, and this variable would study exactly that.

  The construction of this attribute had to be made based on the 'Class' attribute, which indicated if the Lot was VSR or not. However, there may also be some lots that have undergone the treatment but were not identified as such in their aesthetic class. For that reason, this variable may contain some imprecisions.

- ***Error:*** the Error variable was the last one to be designed. A closer inspection to the dataset revealed the most commonly found abnormalities in the observations' values. Table 5.8 shows the identified abnormalities:

Error ID	Error	Desciption
0	-	No error associated to the observation
1	TCA = 0	PeakValue attribute is equal to 0, which should never happen because there is always some variation in the current in the detector, even if not provoked by TCA, but by another compound.
2	Duplicated analyses	PeakValue attribute is exactly equal to the previous record, which is highly unlikely.
3	Analysis Duration < 14	DeltaTime attribute is lower than 14.
4	Analysis Duration > 16	DeltaTime attribute is higher than 16.
5	Incorrect G2	Quality attribute shows that stopper was classified as G2, but PeakValue attribute is not within the thresholds for this category.
6	Incorrect G3	Quality attribute shows that stopper was classified as G3, but PeakValue attribute is not within the thresholds for this category.
7	Integration Error	In some ocasions, due to lack of computational power, for example, the software is unable to perform the integration of the area of the signal of the detector and writes the value '15,5' in the PeakValue attribute.

Table 5.8: Correspondence between 'Error' variable classes and their description

  With this, the 'Error' variable was constructed as a categorical variable whose values correspond to the Error ID column of Table 5.8.

In addition to the creation of these variables, there were some variables whose nomenclature was modified in order for their names to be more clear and explanatory. This way:

- 'AnalTime' was changed to 'AnalysisDate';
- 'PUnit' was changed to 'Module';
- 'DeltaTime' was changed to 'AnalysisDuration'.

All these modifications to the dataset had a clear impact on the dataset structure. There were now 19,792,845 total observations and 14 different attributes. The most noteworthy alterations would be:

- All records present in the dataset correspond to actual measurements of TCA in cork stoppers, not existing signals corresponding to other situations (module start, stop, etc.);

- All categorical variables present the real number of classes, having their values normalized and understandable. Also, a clear reduction of classes is visible across all these attributes, especially for 'Lot', 'Calibre' and 'Class';
- Redundant or irrelevant features have been excluded;
- New variables that may bring useful information to the company were constructed.

Annex C shows in detail the updated and the new variables, including statistical measures, histograms and bar charts for each feature individually.

5.2 Exploration of the clean variables

5.2.1 Relationships between variables

This task consisted of the application of different statistical techniques to obtain information regarding the way variables interact with each other, providing useful information about their relative importance and influence in the results of the chromatographic analyses. The main results obtained were:

- There were several relationships found between the categorical variables using Mutual Information measure:

Variable Pair	Mutual Information
Module | Line	3.31
Module | Lot	2.88
Module | Calibre	1.06
Line | Lot	2.58
Line | Calibre	1.03
Lot | Class	2.86
Lot | Calibre	1.99
Lot | Origin	1.77

Table 5.9: Significant Mutual Information of categorical variables

- Mixing categorical and numerical features and using the Kruskall-Wallis test (α = 0.05) to the differences between medians some curious results were obtained (null hypothesis (H_0): all classes have the same median; H_1: at least one class has a median signigicanltly different from the rest). Though for 'PeakValue' and 'PeakTime' there was no evidence of the existence of a class of a categorical feature that was significantly different from the rest, the same cannot be said about the 'AnalysisDuration' variable. For this variable, there were several categorical attributes where at least one class was significantly different from the rest, as it can be observed in Table 5.10.
- The numerical attributes did not show any correlation among them, with a Pearson Correlation Coefficient always close to 0.

Variable Pair	KW p_{value}
Module | AnalysisDuration	0.23
Line | AnalysisDuration	0.193
Lot | AnalysisDuration	1
Class | AnalysisDuration	1
Calibre | AnalysisDuration	0.38
Origin | AnalysisDuration	0.91
Shift | AnalysisDuration	0.84
VSR | AnalysisDuration	0.30

Table 5.10: Kruskall-Wallis test results where null hypothesis was rejected ($p_{value} > \alpha$)

These results were in accordance with the expected. Among the categorical features, a relationship between the 'Line' and 'Module' features is normal, given that each module belongs to only one line in this data set. Also, because each lot is analysed in a limited number of Modules, it is natural to have a relationship between 'Module' and 'Lot' variables. Then, because each lot's stoppers have the same calibre and aesthetic class the relationship between the two variables and the variable 'Lot' is also quite natural.

On the other hand, the distribution of the 'PeakValue' and 'PeakTime' is the same across all modules, lines, lots, etc. However, for the 'AnalysisDuration' the distribution is different in at least one different population several categorical attributes, namely 'Module' and its associated variable ('Line'), 'Lot' and its associated variables ('Class', 'Calibre', 'Origin' and 'VSR') and 'Shift'. These results show that though the distribution of the 'PeakValue' and 'PeakTime' is independent of the other features as 'Module' and 'Lot'. Contrarily, the distribution of the 'AnalysisDuration' is dependent on almost every other variable. This indicates that possibly some modules or lots are producing taking different duration periods for the chromatographic analysis, leading to the appearance of more registering or reading errors.

Lastly, it can be said that the numerical variables are independent of each other, meaning that the 'PeakTime' and 'AnalysisDuration' variables do not have a significant influence over the 'PeakValue'. This could contradict the previous paragraph regarding the 'AnalysisDuration' feature. Regardless, this may be something for the company to further investigate.

5.2.2 Variable influence over the classification of stoppers

With the information gathered until this point, it was decided to investigate how the different variables influenced the classification of corks. Being the equipment where the detection of TCA occurs, studying how the results vary with the modules was important. Also, the influence of the line and of chamber of the module could also contain relevant information, considering that each line is equivalent to a group of modules, and the chambers are intimately connected to the the functioning of the module. Also, according to the company it would only be natural that each lot would be somewhat different from the others, depending on its origin and on being a VSR lot or not. This way, it was also important to study the influence of the lots, origins and VSR

characteristic over the results of the chromatographic analysis. Annex D provides the graphical information of support of this phase.

Modules:

Being the equipment where the chromatographic analysis takes place, it is important to verify what is the influence that the units have over the classification of the stoppers. In other words, it should be understood how being analysed in a module instead of another influences the chances of a stoppers being classified as good for sale, that is, as G1.

- *Influence on 'Quality':*

  The acceptance of a module is defined as the percentage of corks that the module classifies as G1, which, according to the firm should be above 80% for the generality of the modules. As figure D.1 shows, there are great differences in the tendency to classify corks as G1 among the units. Unit 42 categorized 92.04% of the stoppers it has analysed as G1, whilst for Unit 08 this number drops to 59.27% – a difference of 32.77%.

  Another curious aspect is shown in figure D.4. As observable, the percentage of G3 stoppers is roughly the same for every module. However, there is a clear trade-off between classifying the corks as G1 or as G2. In other words, when a module does not classify a stopper as G1, it will most likely classify it as G2. Even more interesting is that the graph suggests this is a linear trend.

  These are very important and unexpected conclusions, that show that though the modules are all constructed with the same design and materials and are all analysing the same kind of product they present a high level of disparity in their behaviour. Therefore, it can be concluded that the unit where a stopper is analysed has a substantial influence on whether the stopper is classified as G1 or G2. Annex D further supports these conclusions with figures to those mentioned but for G2 and G3.

- *Influence on 'PeakValue':*

  To confirm the thesis that the module greatly influences the classification of a stopper, the same approach was taken for the PeakValue attribute. Being the basis for the classification results obtained in the Quality attribute, the average PeakValue was expected to reflect the same diversity of values across modules. As Figure D.5 confirms it, the average peak value differs significantly across modules.

- *Influence on 'Errors':*

  Lastly, observing the percentage of annotated errors in each module is also useful to have an idea of which modules may be affected by more problems. This way, the percentage of total errors was calculated for each unit, making this analysis independent of the number of stoppers analysed by each module. Figure D.6 presents the graphs of the variation of the total percentage of errors per module. As there observable, once again modules show significant differences in their behaviour. Though the minimum percentage of errors is 4%

there are modules whose percentage of errors in the records is higher than 10%. This means that for these modules, more than 10% of records in the dataset correspond to one of the identified issues identified in the database.

Lines:

Because the modules are grouped into lines, it was also interesting to perform the same analysis done for the modules, but from the perspective of the lines. Figures D.7, D.8, D.9 and D.10 show the graphs referent to this analysis. As it can be observed there, the variability verified for the modules does not hold true for the lines. This suggests that at the moment, the lines are in equilibrium, in the sense that the effect of the modules over the analysis is diluted in every line.

Chambers:

In addition, as a request of the firm, the average 'PeakValue' per chamber was also obtained. This is shown in Figure D.11, where it can be observed that across all modules, the chambers have different sensibilities. That is, some chambers are more sensible to the presence of TCA in the sample. More rigorously, it is not the chamber that is sensible, but the path that connects that chamber to the detector performs better separation of components, better isolating the TCA from other components, which results in a higher 'PeakValue'.

Lot:

The data had 208 different lots represented. In order to make the analysis easier and simpler, a sample of modules was selected, excluding lots responsible for less than 10000 analysis. Also, it was decided that the lots should all come from the most representative origin – Amorim Cork (AC). This way a significant and representative sample of 124 lots was extracted for analysis.

- *Influence on 'Quality':*

  For each of the lots, the percentage of stoppers classified as G1, G2 and G3 was calculated. The acceptance of a lot corresponds to the percentage of stoppers from that lot classified as G1. Figure D.12 shows the percentage of G1 stoppers in each lot of the sample. The figure shows significant differences between the lots. For example, 94.31% of the stoppers from lot '4841307' was classified as G1, while lot '596001' had an acceptance of 62.58%, which is makes a difference of 31.73%.

  Additionally, observing Figure D.15 bellow, that shows the stacked percentages of G1, G2 and G3 per lot provide additional information: there is a clear trade-off between classifying the stoppers as G1 or G2. In other words, while the percentage of G3 is relatively constant (2 to 6%) across lots, the same cannot be said about G1 and G2. In fact, the reduction of the percentage of G1 stoppers per lot is inversely matched with a rise in the percentage of G2 stoppers. In addition, this approximates to a linear trend, meaning that every lot is different from the others and there cannot be drawn groups of lots. (Annex D presents graphs similar to D.12 but for classes G2 and G3).

- *Influence on 'PeakValue':*

Being the variable that serves as the basis for the classification into the three groups, it is also important to verify the variation of this feature with the lot. For that, Figure D.16 shows the average 'PeakValue' per lot. Once again, the lots display great differences between each other, with some lots present average PeakValues even higher than 12 mV.s (recall that 6 mV.s is the threshold separating G1 and G2). However, it must be referred that this average may be biased by those observations where the PeakValue reaches the tens of thousands of mV.s. In other words, if one of the lots has only a handful of such records, its average PeakValue will increase significantly, which may be the case here.

Origin:

Representing the connection between the different lots in terms of where they came from, the influence of this variable over the classification of the stoppers would be useful to show that certain lots should be expected to have lower acceptance levels. Figure D.17 shows the percentage of G1 stoppers obtained from each origin. However, this time it is concluded that origin of the lots does not have influence on the acceptance of a cork. In other words, the acceptance level of a lot cannot be assumed to be higher or lower than the average apriori. The exceptions are the origins Laboratory (L) and NDTG3 which are in fact special kinds of lots, that were already known to actually be different from the rest.

VSR:

Finally, lots can be divided in accordance to being VSR or not. As referred, the newer lots that come from inside the company all have the VSR treatment prior to arriving to NDtech, reason why the first control reported in chapter 3 was discontinued for those lots. Therefore, it would be interesting for the company to observe the impact of the treatment in the classification of the corks as G1. Figure D.18 compares the acceptance level for VSR lots and non-VSR lots. As observable, VSR lots have an acceptance of 86% while non-VSR lots have an acceptance of only 80%. This way, it can be concluded that the VSR treatment is producing positive results, eliminating TCA from cork stoppers and increasing their chance of being accepted for sale. Though a 6% difference may not seem that relevant, it must be considered that the division of lots in VSR and non-VSR was not perfect, existing the possibility of some VSR lots having been included in the non-VSR group. For this reason, the difference can potentially be higher in reality.

5.2.3 Selection of the sample

To investigate how to select the sample, the graph in Figure H.1 was constructed. It shows how different modules evaluated a sample of lots. In other words, it shows the acceptance of the lots from the perspective of each module.

As observable in the figure, each lot had very different acceptance levels depending on the module. The first lot of the graph ('26001315044'), for example had an acceptance level that ranged between more than 95% and around 65% depending on the module. With this, it is concluded that the Module is truly a great independent influence on the classification of the stoppers.

Therefore, the sample to extract to perform pattern detection should be representative of the diversity of the modules.

To do this, ranking of the modules was constructed in the following steps:

1. For each lot a ranking of the units that have analysed its stoppers was generated based on the percentage of G1 stoppers each unit classified;

2. Then, because each lot was analysed by a different number of units, each rank had to be weighted. For example, a fourth place in a ranking of five modules can not have the same importance as a fourth place in a ranking of 20 modules. This way, each position in each ranking was divided by the number of units in the respective ranking;

3. Finally, for each module, the average weighted rank was calculating, producing the final score, over which the final module rank was calculated, which is shown in Table H.1.

As it can be observed in that table, units 08 and 14 were the modules at the opposite ends of the ranking, with the first having the lowest score, and thus the lowest propensity to classify a random stopper as G1 and the latter with the highest score, and thus having the highest propensity to classify a random stopper as G1. Neither of the cases can be considered better than the other because Unit 08, with less probability of categorizing corks as G1 could be producing many false positives, i.e., stoppers that are falsely classified as having an unacceptable level of TCA concentration. On the other hand, Unit 14 could represent the opposite case, producing many false negatives, i.e., stoppers falsely classified as not having too much TCA. This way, with the help of Figure H.2, 10 modules were chosen to perform the investigation of patterns.

The 10 modules are divided in three groups:

- Modules with the lowest score – 08, 61, 29 and 62;
- Modules with the highest score – 14, 33 and 27;
- Modules closest to the average score (0.568) – 43, 11 and 17.

This way, the chosen sample of modules includes the modules that are potentially producing the most false negatives or false positives, or producing the least false positives and false negatives. The possibility of comparison of patterns within a group and between groups is also an advantage of this sampling option.

5.3 Detection of anomalous patterns

5.3.1 Clustering approach

Utilizing the clustering approach described in the previous chapter some interesting results were obtained. Table E.1 in Annex E presents a summary of the results obtained for each module of the sample, with each of the three defined frame sizes.

Analysing the table, it is not clear what is the effect that a higher or lower score unit has on the results. Indeed, there can not be made a clear distinction of pattern regarding the eps value for example. For every type group of units, the eps value varies differently according to the frame size and to the unit, not existing clear similarities between units of similar score, nor clear differences between units of different types. The number of clusters, the percentage of normal data and the percentage of examples in the second biggest cluster (the biggest anomalous cluster) are also random, once more, not existing clear similarities or differences within or between the groups of units.

Nonetheless, when the anomalies found are observed, there can be drawn some conclusions regarding the effect of the frame size and the score of the unit. First, looking at the type of anomalous patterns, there are clear groups of patterns found across all the analysed units. The most common identified patterns were:

- 8G2 Control patterns:

  Figure E.1 shows a set of frames that constitute a cluster. The frames are separated by two empty (white) columns, and in this case, the frame size is four cycles. This cluster shows the pattern of an 8G2 control, where an operator injects a TCA concentration in the eight chambers consecutively. As observable, the frames present entire cycles where the 'PeakValue' values of the eight chambers are clearly higher than in other cycles (a control does not necessarily start in chamber one). Also, because the colour scale is logarithmic, it is perceivable that the controls have much higher values than regular analysis. Using a frame size of 8 cycles, similar patterns were found, but not for a frame size of 16.

- 4G2 Control patterns:

  For this case, Figure E.2 shows several 4G2 controls, where it is clearly visible that 4 consecutive chambers have higher 'PeakValues' compared to the rest of the analyses in the frame. In this case, this pattern was identified with every frame size

- Integration error pattern:

  This is another pattern that was found in multiple units, not being restricted to just one group of units. Its shows entire frames populated with 'PeakValue' equal to 15.5 mV. This corresponds exactly to the previous annotated error of integration that happens when the software of the unit does not have enough computational power to calculate the area of the peak registered by the detector. Apparently, when this occurs, the modules suffer from this error continuously. This way, whenever an integration error occur the next analyses will all be affected. Therefore, the unit should be intervened or restarted as soon as one integration error occurs. This pattern may be observed in Figure E.3.

- Isolated high peak pattern:

  These patterns show situations were in the middle of a frame, where the 'PeakValue' is constantly low, suddenly an observation appears with a much higher 'PeakValue'. The example

shown for this situation in Figure E.4, present an example for one chamber. However, these patterns were found for multiple chambers in every module, being present across the entire sample. It is not necessarily an anomaly, since it is possible that a stopper does have a high TCA value. Therefore, though this is a pattern that can be identified as different from the normal situation, it is not necessarily an anomalous phenomenon or an error in the chromatographic analyses.

- Cyclic Carry-Over patterns:

  Patterns compliant with the description of Cyclic Carry-Over were also identified in some of the units. As observable in the figures above, for multiple cycles, the same chamber presents consistently high TCA values. This is in accordance with the fact that this type of carry-over is provoked mainly by debris inside the chamber which are not detected by the technicians immediately, and thus, remain inside the chamber for a number of cycles. These patterns were not detected in the medium score modules. Figure E.5 presents the graph of respect to this pattern.

- Chamber Carry-Over and Cyclic Carry-Over pattern:

  In the case of Chamber Carry-Over, the identified patterns were always associated with Cyclic Carry-Over as well, as shown in Figure E.6. This demonstrates that while for the Cyclic CO issue, the cause is presumably debris in the chamber, for the Chamber CO the debris is located along the path that connects the chamber to the detector and is shared by each two columns. This way, whenever Chamber CO occurs, apparently, so it will occur Cyclic CO.

These results show that any pattern of the identified above can be found with any frame size in all groups of modules, whether they had a high, medium, or low score in the module ranking. However, mainly for the 4G2 and 8G2 Control patterns, the increase of the size of the window diluted the effect of the control and the statistics extracted from the frames did not differ from normal frames, leading to a higher scarcity of this type of pattern when the frame size was higher. This way, it would be preferable to use smaller frame sizes in this type of approach.

Furthermore, it was noticed that both of the control patterns identified were more easily detectable in medium score units, regardless of the frame size. In other words, though for those modules, the other types of anomalous were also identifiable, the large majority of patterns in such units corresponded to control patterns. Contrarily, both low and high score modules had a higher prevalence of other types of modules such as carry overs or the isolated peaks.

5.3.2 Secondary approaches

5.3.2.1 PeakValue variation signal

For the first of these approaches, the approach was performed in the samples of modules 08, 43 and 14. The histogram obtained in the three cases was nearly the same. Annex F, presents the

histogram obtained for unit 08 (Figure F.1).

As observable, the majority classes are those whose value are only ‘1’ and ‘-1’. This was expected because there were not many ‘0’ and ‘2’ since missing records and repeated values for a chamber in consecutive cycles were minorities of the data. However, it is interesting to notice the difference between the frequency of sequences ‘1-1’ and ‘-11’ and the frequency of sequences ‘11’ and ‘-1-1’. In fact, the first two occur with a frequency that is almost the double (30%) of the frequency of the last two sequences (17%). In other words, it is much more likely that the signal of a chamber changes the signal of its variation than it keeps it. This means that whenever the value of the peak value rises when compared to a previous cycle, in the next cycle it will most likely fall and vice-versa.

Also, this shows that each chamber’s values are constantly rising and dropping consecutively. This way, it can be said that the signal of the difference between the peak value of consecutive analyses in the same chamber is not random. If it was, it should be seen a similar frequency for the four most common sequences, ‘11’, ‘-1-1’, ‘1-1’ and ‘-11’. As it is, clearly the signature pattern of the units is consecutive rises and falls in the peakvalues of the eight chambers.

As a proof of this, Figure F.2 shows the histogram created for a segment of unit 14 with frame sizes of 4 cycles. As it is observable, the most common sequences are those where the value of the signal is consecutively alternated or almost that way.

5.3.2.2 Error patterns

For the other secondary approach, which used the ’Error’ variable, the histograms obtained were once more nearly the same for the three modules (08, 43 an 04) where this was performed. Annex G shows the case of unit 14, where it is observable that the most common type of cycle is that where the values of the ‘Error’ variable are always 0, which corresponds to the case where no error was identified. However, looking at the following classes it is noticeable that by far the most common type of cycle is that where error 9 is immediately followed by error 4. In other words, a missing record is followed by a record with an ’AnalysisDuration’ higher than 16 seconds. This shows it paid-off noting the missing records with the value ‘9’. Also, this means that the modules present a common pattern that indicates that whenever an observation presents an analysis duration time higher than 16 seconds the system probably has failed to register the previous observation in the database.

Chapter 6

Conclusion and future work

The main objective of the project was to perform a study on the anomalous patterns present in the data generated by the company's technology. With that, useful information could be provided to the company, which can use it as support for future decision-making. In a final instance, those decisions should converge to the increase of the company's equipment reliability, consequently reducing costs. This was achieved, not only providing relevant information regarding the patterns but also gathering additional information regarding other issues such as data quality.

6.1 Main conclusions

The results presented in the previous chapter expose many of the peculiarities of NDtech. Because this was the first time this data was carefully and thoroughly analysed, some steps of the project were quite challenging, but also many conclusions were taken because of that.

In the first place, the initial analysis to the dataset revealed many issues with its quality. It was discovered that some modules are not connected to the database, either always or occasionally, which should be resolved in order for the firm to collect data from absolutely every unit they have.

Many impossible values such as fictitious modules, line, lots, classes, calibres, etc. were found to be polluting the database with information that is inaccurate and unrealistic. These are mainly caused by the form of introduction of such information in the system, which is currently done manually and without any standardization, leaving space for many human errors and inconsistencies between the representation each employee uses for the same value. Additionally, it was demonstrated the redundancy of having two different variables portraying the line of the module, especially considering that one of them had close to 50% of damaged/incorrect observations. Furthermore, the inclusion of new variables in the dataset, such as those three created, 'Origin', 'VSR' and 'Errors' revealed helpful to help the firm understanding the impacts of those characteristics in the quality of their stoppers, thus its permanent addition to the set of variables of the database is recommended. Lastly, it was concluded that one of the main issues of the database is related with missing records. Several chromatographic analysis did not get registered in the database, which represents a loss of information, and it has consequences on the registry of the following

chromatographic analysis, which get registered with values that possibly do not correspond to reality.

These results point to the need of a better procedure for the introduction of many variables in the system. Using QR codes or bar codes could be a solution to the problem. The reading of these codes with a code scanner would automatically introduce the correct values in the system, namely for the 'Lot', 'Class' and 'Calibre' variables. Another possibility is to force the introduction of these variables in the system before the initialization of the units, which would prevent the technicians of forgetting to introduce the required parameters, whether manually or using codes.

In addition, the presented preprocessing actions can serve as a framework for future analyses. Many of the required steps have already been established in this project and it would be easy to follow the same procedures. Particularly, the creation of the variable 'Error' may be very useful in identifying potential issues of the data and of the modules as well.

Furthermore, the exploratory analysis of the characteristics of the features of the dataset revealed important facts that are of the highest value to the company. Clearly, the population of units is the biggest factor of influence in the classification of a stopper. With the ranking of modules, the company can now assess which modules behave abnormally and take special attention to these.Also, the chambers were also found to present differences among them. Some are more sensible than others. Namely, chambers 1 and 5, which share the same analytical column, are more sensible than the rest and have higher average peak values. This may indicate that the design of the units may be leading to a higher probability of accumulation of debris in the column of these chambers, resulting in more detected TCA.

Regarding the actual detection of patterns in the data, many interesting results were also collected. The clustering approach revealed potential in detecting anomalies in the data. Unlike the current rule-based implementation, the clustering technique was able to distinguish data produced by the controls, which is useful because these change the dynamics and context of the subsequent chromatographic analyses. The currently implemented rules are oblivious to this. Hence, by using the identification of the controls performed by the clustering algorithm, the rules can be adapted to take into account the change in the dynamics of the module.

The clustering approach was also able to identify both types of carry over, discovering that the Chamber Carry over issue is always accompanied by Cyclic carry over. Furthermore, it has demonstrated the importance of intervening a module as soon as an integration error is detected, otherwise multiple chromatographic analysis will be performed in vain. On the other side, creating clusters where there is only one chamber that appeared with an unusually higher TCA value may be a downside, because this may not be a problem of the unit, but an actual TCA value of the cork.

Therefore, using an approach based on clustering to detect anomalies in a real-time context can be useful, to help guide other approach, namely, the current rule-based implementation.

The two secondary experiments have also uncovered important facts regarding the GC units' functioning. First, with no apparent reason for it, it seems like the units are forcing to keep the TCA values constant, by constantly compensating rises in the peak value of a chamber with a drop in the subsequent cycle. Second, it was demonstrated that the most common error in the register

of data corresponds to missing observations which are then followed by observations with high analysis duration times. Therefore, it can be assumed that whenever a data point appears with an analysis duration higher than 16 seconds, then, almost certainly, the previous chromatographic analysis failed to be registered in the database.

All these results contain information of high value to the company, which can now assess it and take the necessary actions to improve the registration of data, and the functioning of the chromatographic units and NDtech as whole. Ultimately, this will lead to a higher reliability of the modules, and consequently to fewer financial losses to the company, thus increasing the value it generates.

The methodology itself proved to be exhaustive and adequate for the first steps of the project. Namely, for the preprocessing, and exploration of the clean dataset, the results obtained show it was a good approach. Nonetheless, in the preprocessing stage particularly, it was also inefficient, taking a long time to conduct the entirety of the process. Regarding the application of the anomaly detection techniques, some interesting results were obtained, though it can be argued if clustering was the best choice. In fact, other techniques such as VSM or Isolation Forrest could have been more efficient in separating anomalous data from normal instances. However, the fact that the clustering approach is also able to differentiate between types of anomalies is its main advantage.

6.2 Future work

Though having obtained information of interest to the company, and having performed an important analysis to the current situation of the data produced by NDtech, this work can still be further developed in the future.

In future analyses to the data, it would be advantageous to have a framework of preprocessing steps already predefined. As described in this document, preprocessing steps took the majority of the time available for the execution of this work, which compromised a deeper subsequent analysis. The steps of the preprocessing framework reported in this dissertation can be used to create such guideline.

A normalization of the process of introduction of many variables in the system has also been shown to be required. Several measures can be adopted such as those suggested in Chapter 6, to avoid situations caused by the forgetfulness of the employees on registering or updating several parameters of the system.

Further developing the clustering approach presented in this project would also be of interest to the company. Namely, creating an implementation to detect anomalies in real-time would allow at least in a first phase to support the detection of anomalies performed by the currently implemented rule-based anomaly detection. For that, further studying of which is the best frame size for this data would allow for a more concrete and better-defined approach. Also, extending the analysis performed in the sample of ten chromatographic units to the rest of the population would be useful to confirm or refute many of the conclusions that were here drawn.

Lastly, many of the anomaly detection techniques mentioned in the second chapter can also be used to detect anomalies in the data. In the work developed in this project, it was opted to work with a clustering algorithm, (DBSCAN) because of the main advantage of identifying different groups of anomalies, which some other techniques cannot do. This is a useful advantage for an exploratory analysis such as the one here presented, indicating what are the most important anomalies to consider. However other techniques may also be useful to explore NDtech's data. Namely, supervised techniques can be employed to detect specific anomalies, such as those that have been identified in this document. For example, Naive Bayes algorithms present the advantage of low computational complexity and Decision Trees techniques are often very accurate. Using regression models is another option to consider because of its simplicity of application.

Bibliography

Agrawal, S. and J. Agrawal (2015). Survey on anomaly detection using data mining techniques. In *Procedia Computer Science*, Volume 60, pp. 708–713. Elsevier B.V., Netherlands.

Alasadi, S. A. and W. S. Bhaya (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences 12*(16), 4102–4107.

Alhamdi, F. M. (2020). Role of packaging in consumer buying behavior. *Management Science Letters 10*(6), 1191–1196.

Altin, C. and O. Er (2016, 08). Comparison of different time and frequency domain feature extraction methods on elbow gesture's emg. *European Journal of Interdisciplinary Studies 5*, 35.

Amorim Cork (2020a). A cortiÇa e o vinho - cortiÇa vs vedantes artificiais. https://www.amorimcork.com/pt/a-cortica-e-o-vinho/cortica-vs-vedantes-artificiais/. [Online; accessed 25-October-2020].

Amorim Cork (2020b). Portfolio. https://www.amorimcork.com/en/portfolio/d/ndtech/231/.

APCOR (2019a). boletim_estatistico_apcor_2019. Technical report, APCOR.

APCOR (2019b). Relatório Anual de Caracterização da Situação Económico-Financeira das Empresas do Setor da Cortiça. Technical report.

Basu, S. and M. Meckesheimer (2007, feb). Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems 11*(2).

Blázquez-García, A., A. Conde, U. Mori, and J. A. Lozano (2020, feb). A review on outlier/anomaly detection in time series data.

Caesarendra, W. and T. Tjahjowidodo (2017, dec). A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing.

Chandola, V., B. Arindam, and K. Vipin (2007). Outlier Detection-A Survey. Technical report.

Chandola, V., A. Banerjee, and V. Kumar (2009, 7). Anomaly detection. *ACM Computing Surveys 41*.

Chandola, V., A. Banerjee, and V. Kumar (2012, may). Anomaly detection for discrete sequences: A survey.

Cook, A. A., G. Misirli, and Z. Fan (2020, jul). Anomaly Detection for IoT Time-Series Data: A Survey.

Cork Quality Council (2020a). Cork forests. https://www.corkqc.com/pages/cork-forests.

Cork Quality Council (2020b). Cork physical attributes. https://www.corkqc.com/pages/cork-physical-attributes.

Corticeira Amorim (2020a). Corporate milestones. https://www.amorim.com/en/who-are-we/corporate-milestones/.

Corticeira Amorim (2020b). Key characteristics. https://www.amorim.com/a-cortica/caracteristicas/.

Corticeira Amorim (2020c). Raw material and industrial process. https://www.amorimcork.com/en/cork-and-wine/raw-material-and-production-process/.

Das, K., R. N. Behera, B. Tech, and A. Professor (2017). A survey on machine learning: Concept, algorithms and applications machine learning view project international journal of innovative research in computer and communication engineering a survey on machine learning: Concept, algorithms and applications. *Article in International Journal of Innovative Research in Computer and Communication Engineering*.

Delloyd's Lab-Tech Chemistry (n.d.). Capillary column gas chromatography. http://delloyd.50megs.com/MOBILE/GC.html.

EY (2019). Pegada de carbono da rolha de cortiça natural. Technical report.

Guerra-Hernández, A., R. Mondragón-Becerra, and N. Cruz-Ramírez (2008, 01). Explorations of the bdi multi-agent support for the knowledge discovery in databases process. *Research in Computing Science 39*, 221–238.

Harris, D. (2008). *Análise química quantitativa*. LTC.

Hayashi, C. (1998). What is Data Science ? Fundamental Concepts and a Heuristic Example. In *Data science, classification, and related methods*, pp. 40–51. Springer.

He, Z., X. Xu, and S. Deng (2003, June). Discovering cluster-based local outliers. *Pattern Recogn. Lett. 24*(9–10), 1641–1650.

Import.io (2017). Data mining vs. machine learning: What's the difference? https://www.import.io/post/data-mining-machine-learning-difference/.

International Trade Statistics (2020). List of exporters for the selected product - product: 45 cork and articles of cork. https://www.trademap.org/tradestat/Country$_{S}elProduct_{T}S.aspx?nvpm = 1$

Liu, T., H. Gao, and J. Wu (2020). Review of Outlier Detection Algorithms Based on Grain Storage Temperature Data. In *Proceedings of 2020 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2020*, pp. 1045–1048. IEEE, Piscataway, NJ, USA.

Men, H., Y. Jiao, Y. Shi, F. Gong, Y. Chen, H. Fang, and J. Liu (2018). Odor fingerprint analysis using feature mining method based on olfactory sensory evaluation. *Sensors (Switzerland) 18*(10).

Ramírez-Gallego, S., B. Krawczyk, S. García, M. Woźniak, and F. Herrera (2017). A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing 239*, 39–57.

Reynolds, D., I. Rahman, S. Bernard, and A. Holbrook (2018, sep). What effect does wine bottle closure type have on perceptions of wine attributes? *International Journal of Hospitality Management 75*, 171–178.

Schmitt, P. (2017). Wine closures: The facts. https://www.thedrinksbusiness.com/2017/02/wine-closures-the-facts/.

Scott, R. P. (2003). Chrom-ed book series - book 2. *Liquid Chromatography*.

Sefton, M. A. and R. F. Simpson (2005). Compounds causing cork taint and the factors affecting their transfer from natural cork closures to wine – a review. *Australian Journal of Grape and Wine Research 11*(2), 226–240.

Shanthamallu, U. S., A. Spanias, C. Tepedelenlioglu, and M. Stanley (2018). A brief survey of machine learning methods and their sensor and IoT applications. In *2017 8th International Conference on Information, Intelligence, Systems and Applications, IISA 2017*, Volume 2018-Janua, pp. 1–8.

Silayoi, P. and M. Speece (2007). The importance of packaging attributes: A conjoint analysis approach. *European Journal of Marketing 41*(11-12), 1495–1517.

Smith, R., A. Bivens, M. Embrechts, C. Palagiri, and B. Szymanski (2002, 01). Clustering approaches for anomaly based intrusion detection. *Proceedings of Intelligent Engineering Systems Through Artificial Neural Networks*, 579–584.

Yang, H. (2013). Data Preprocessing-Chapter 3. *citeseerx*.

Appendix A

Bar charts and histogramms of raw data variables

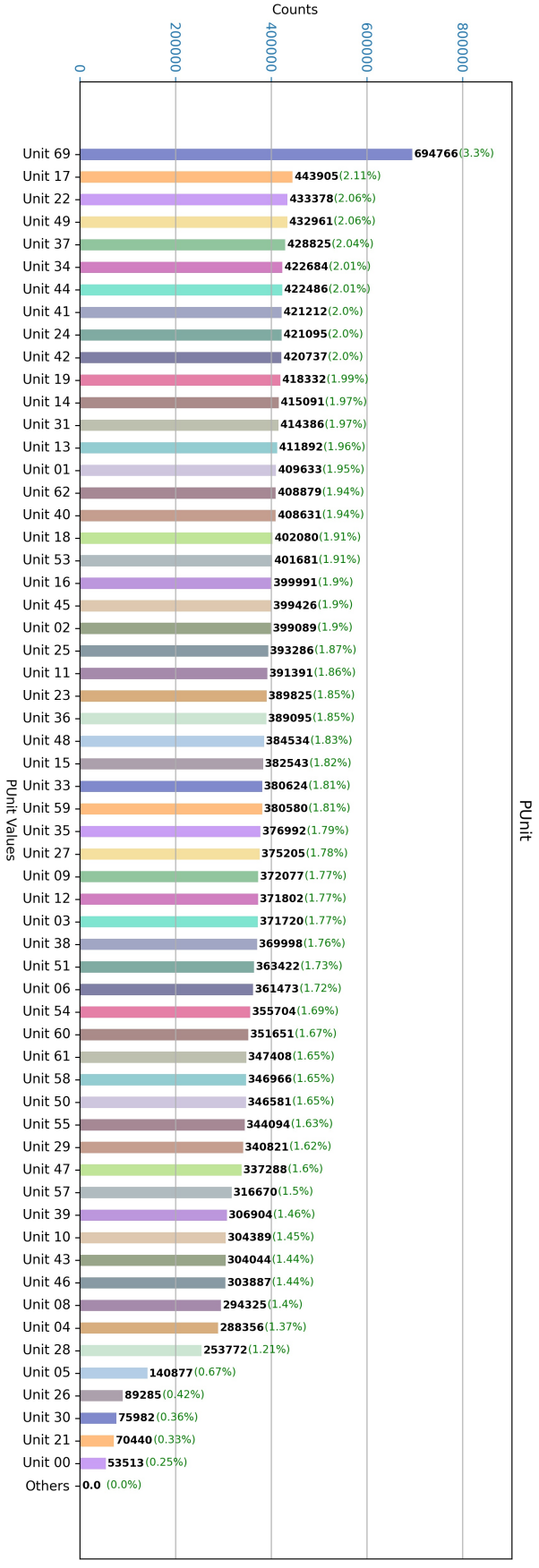


Figure A.1: Bar chart of raw 'Punit' variable

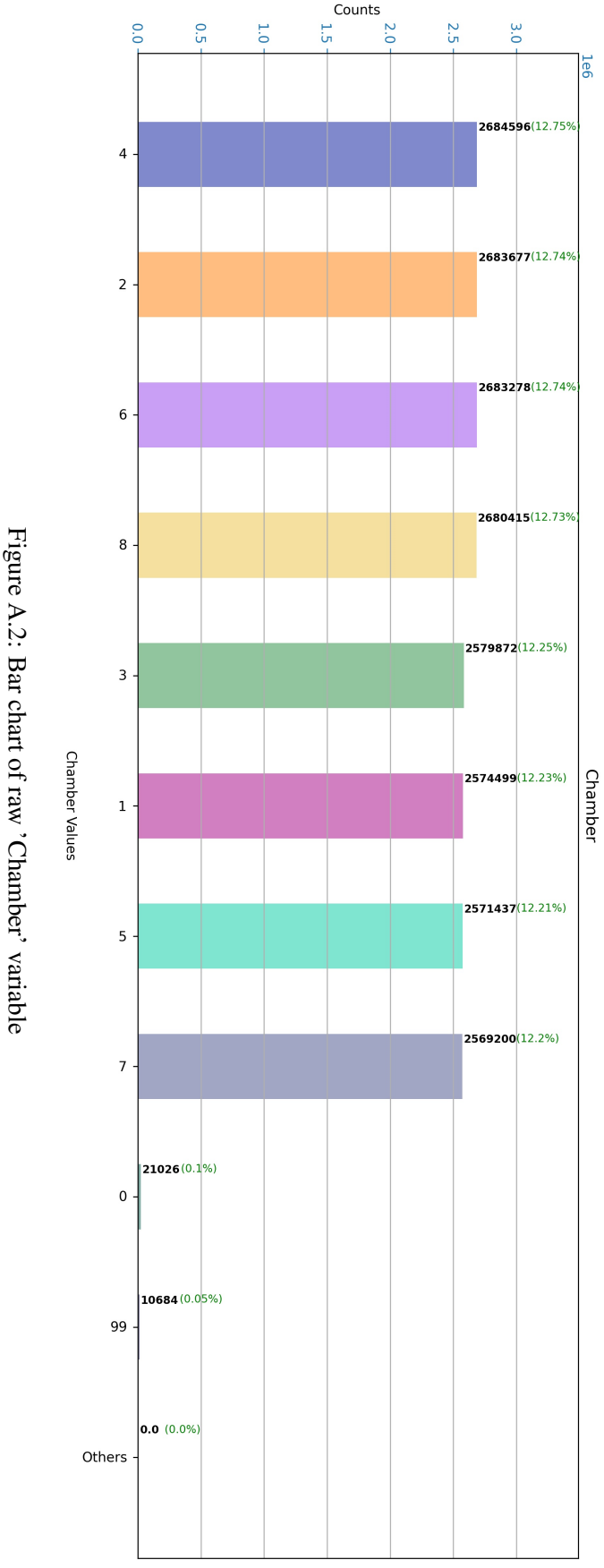


Figure A.2: Bar chart of raw 'Chamber' variable

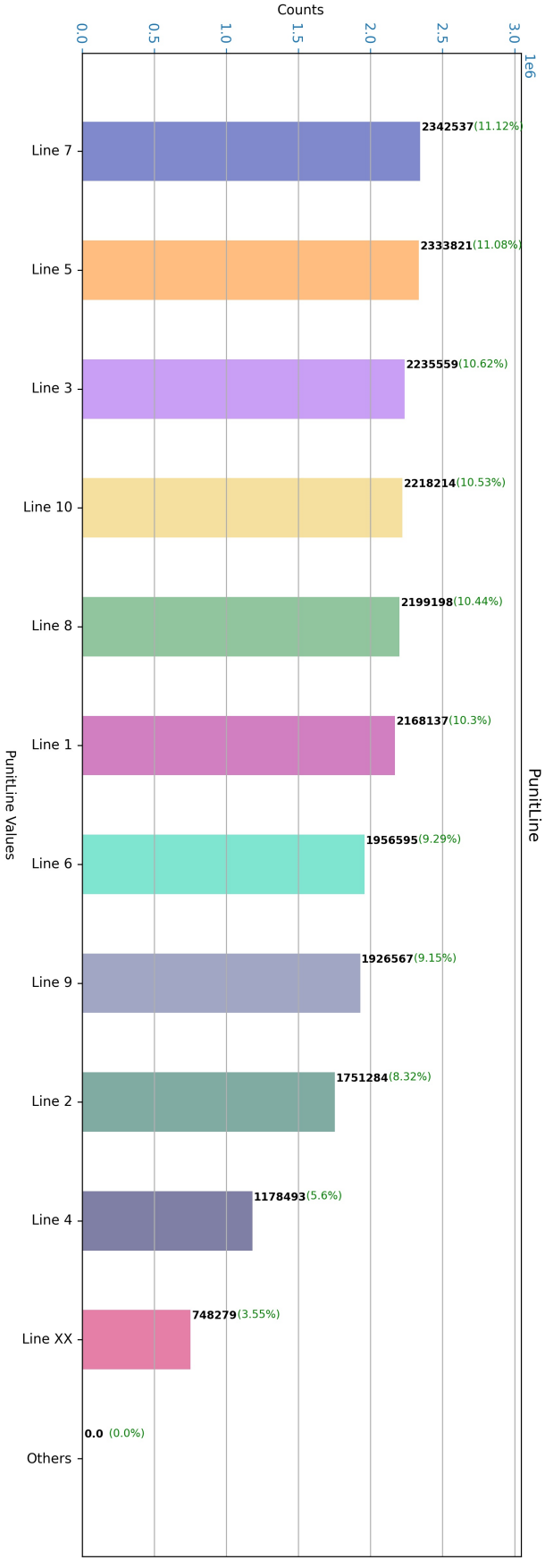


Figure A.3: Bar chart of raw 'PunitLine' variable

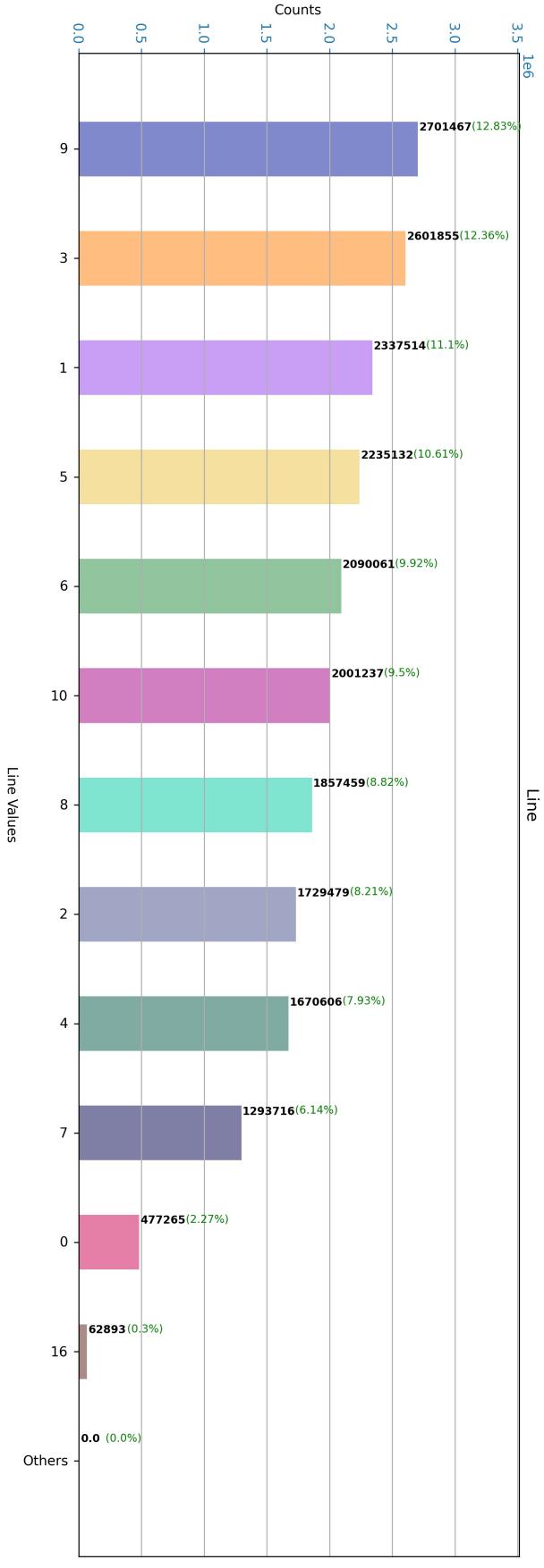


Figure A.4: Bar chart of raw 'Line' variable

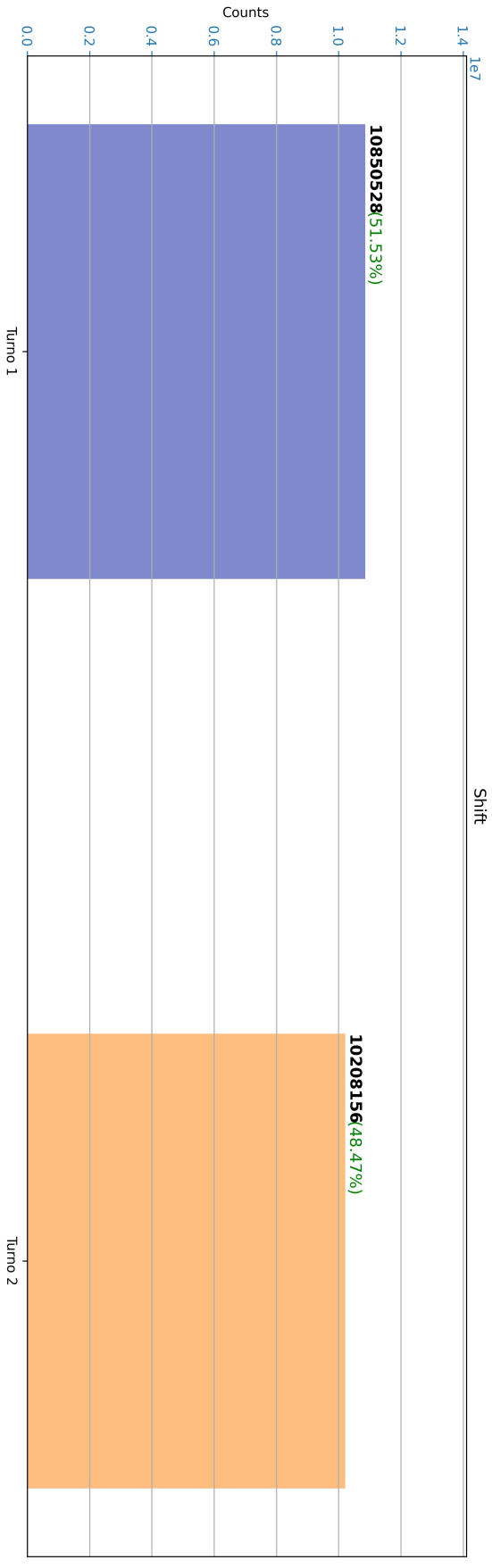


Figure A.5: Bar chart of raw 'Shift' variable

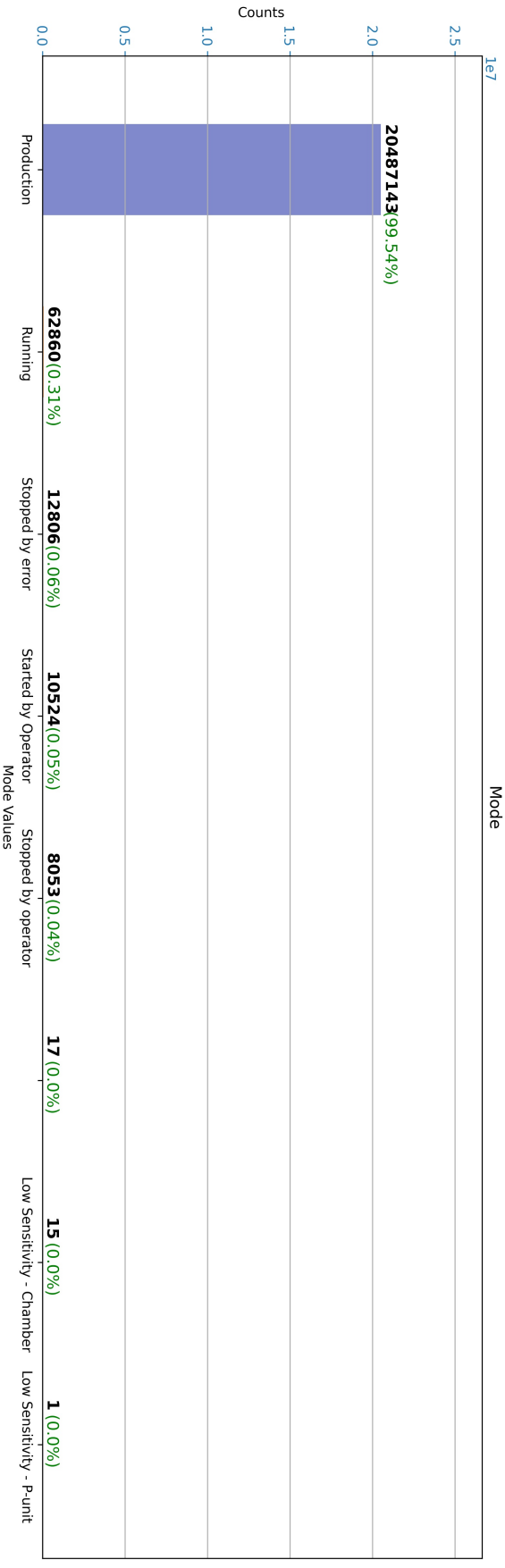


Figure A.6: Bar chart of raw 'Mode' variable

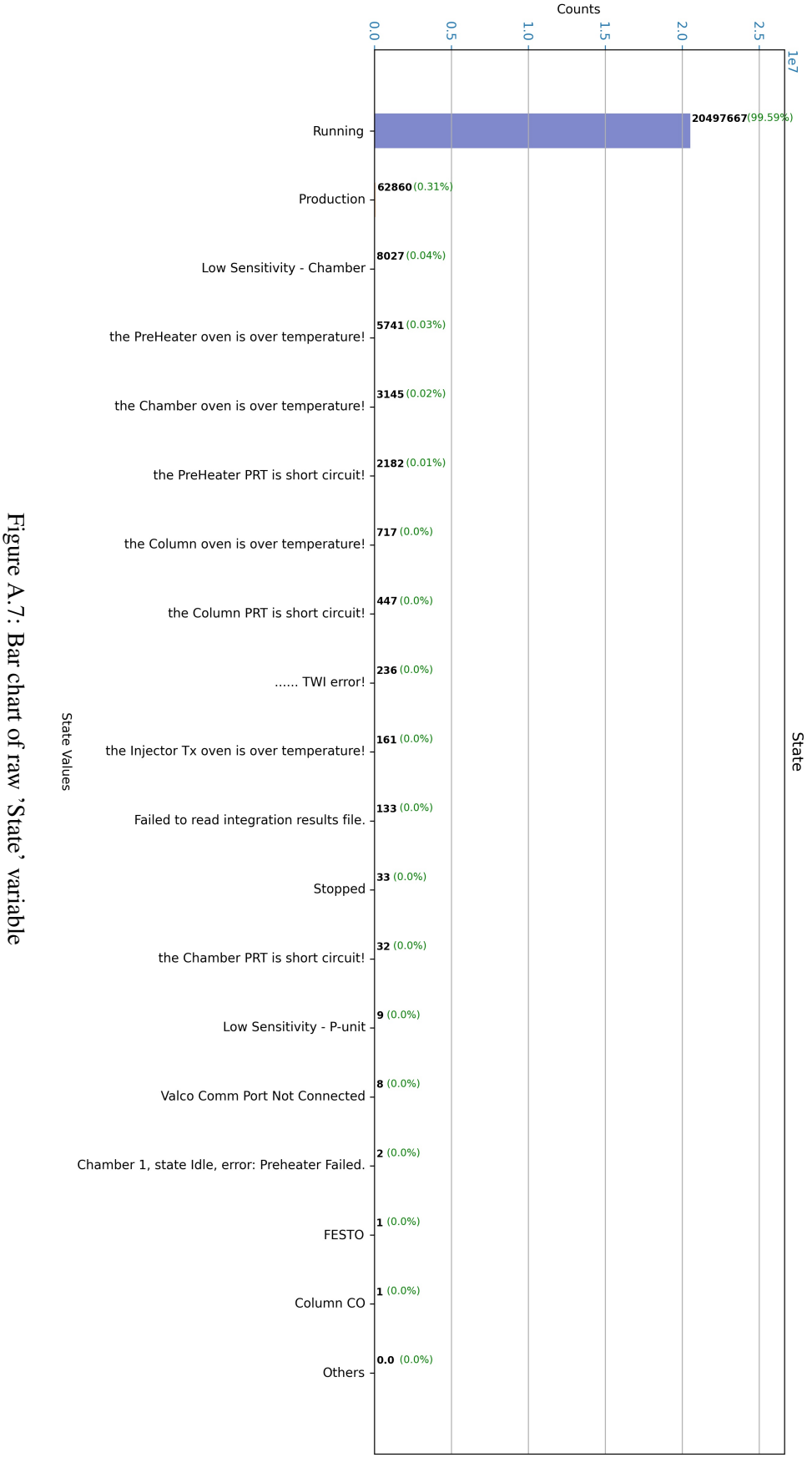


Figure A.7: Bar chart of raw 'State' variable

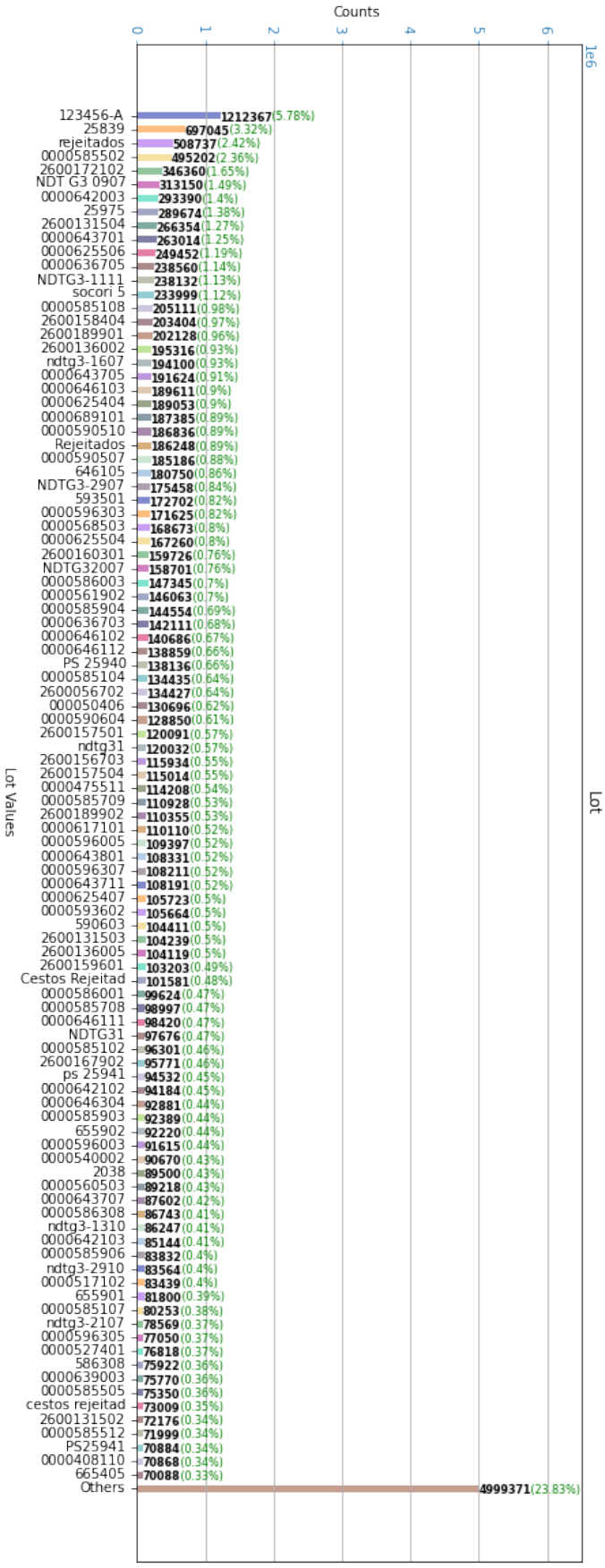


Figure A.8: Bar chart of raw 'Lot' variable

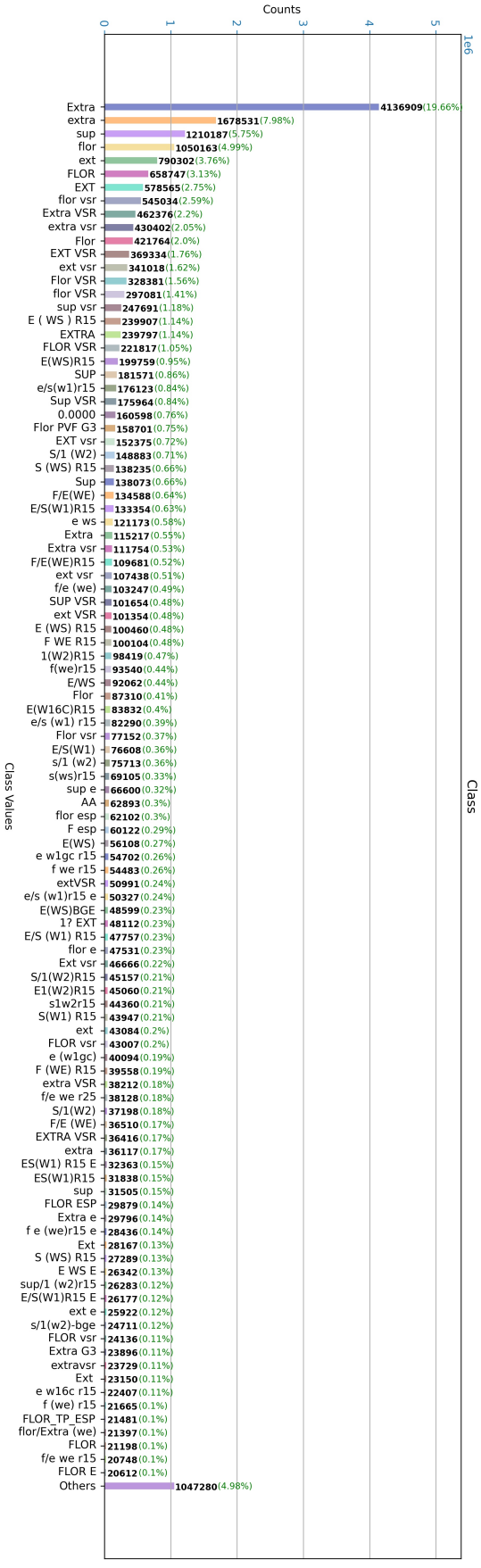


Figure A.9: Bar chart of raw 'Class' variable



Figure A.10: Bar chart of raw 'Calibre' variable

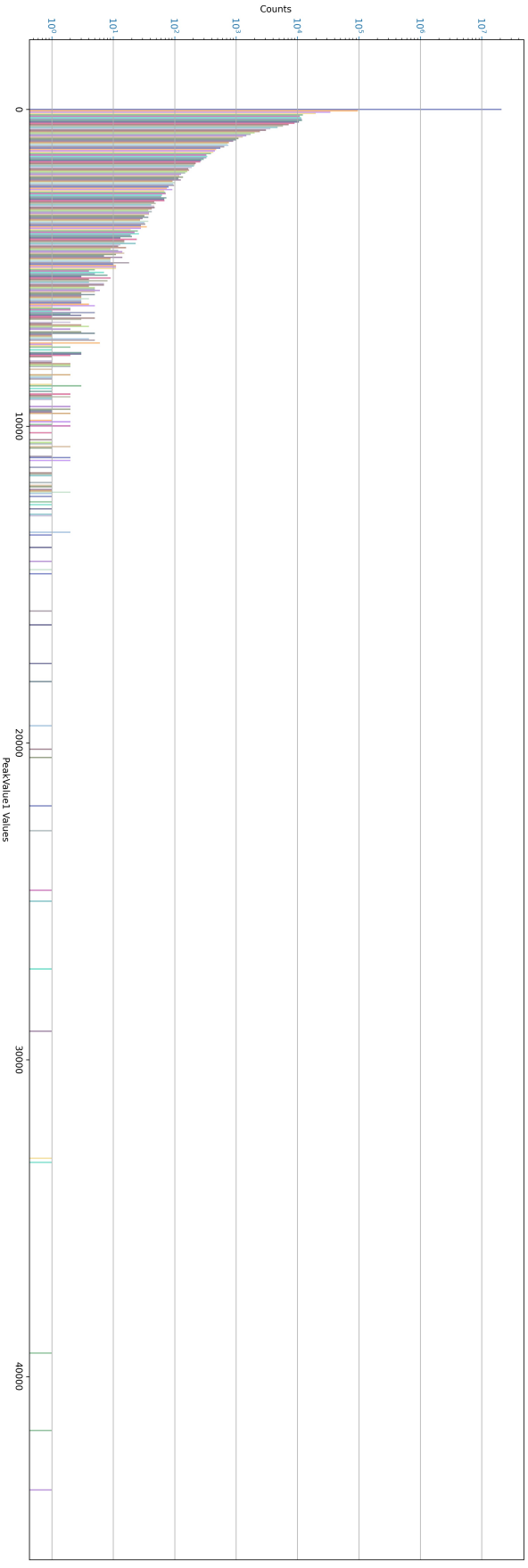


Figure A.11: Histogram of raw 'Peakvalue' variable - all values

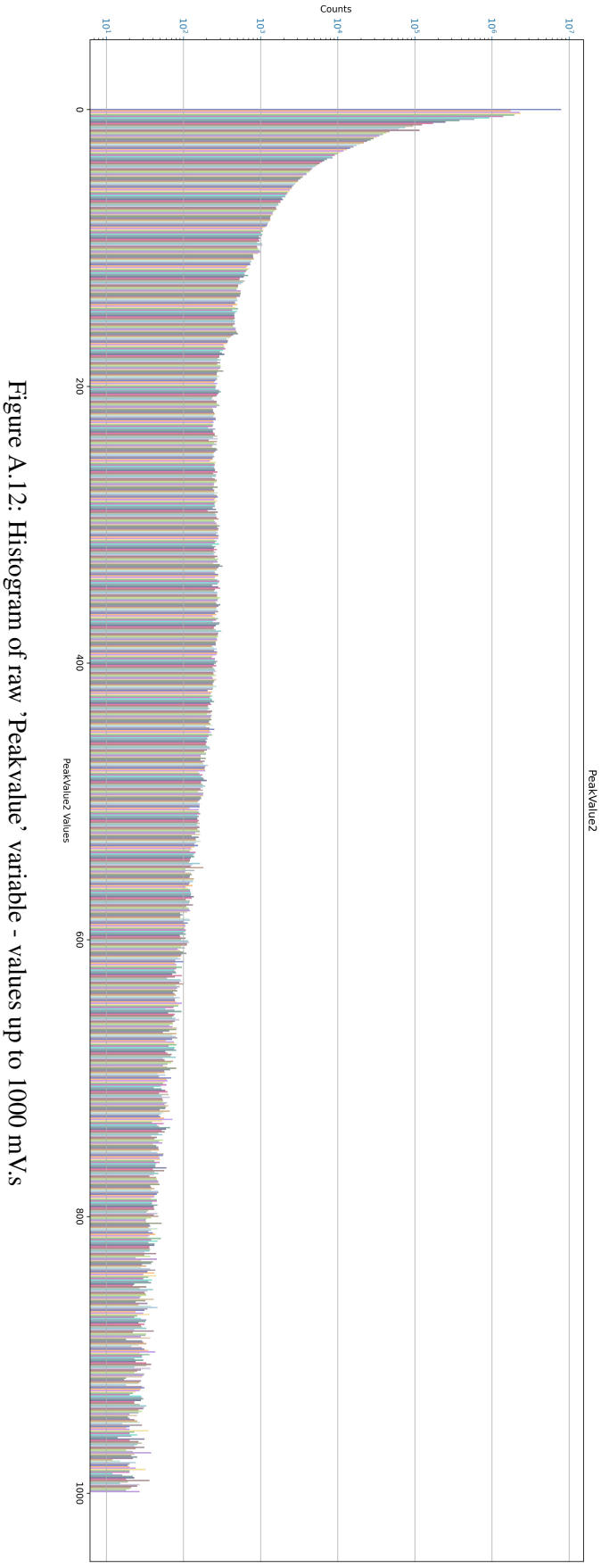


Figure A.12: Histogram of raw 'Peakvalue' variable - values up to 1000 mV.s

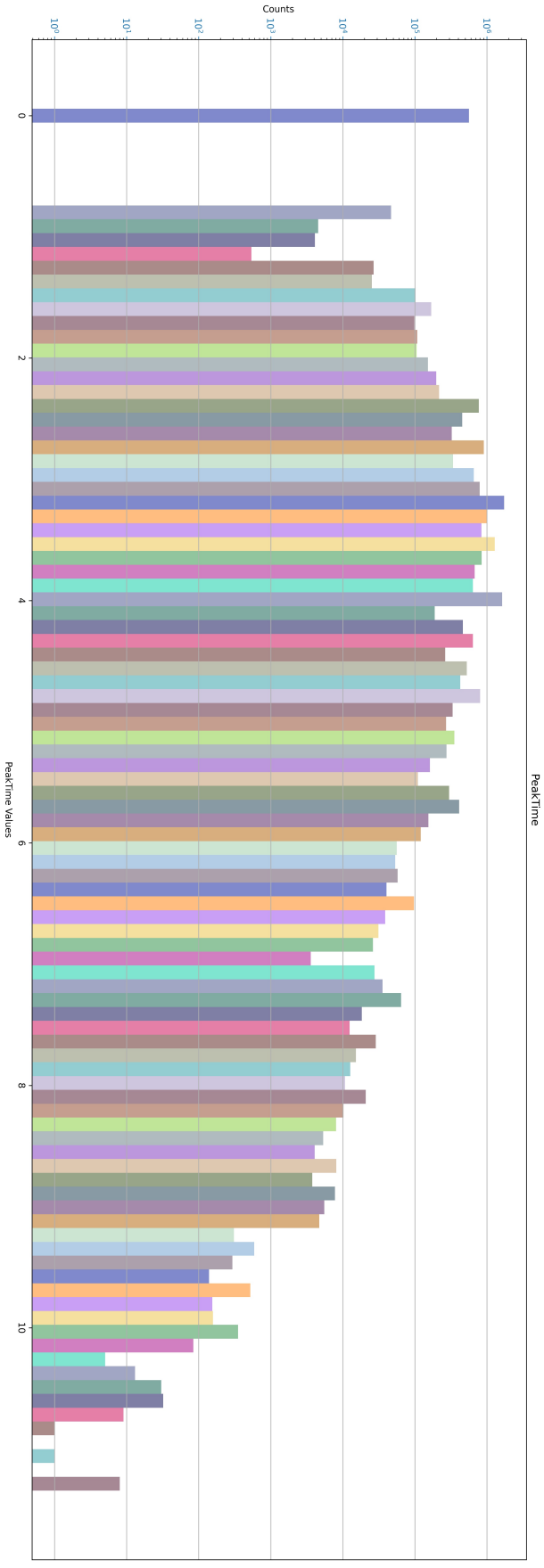


Figure A.13: Histogram of raw 'PeakTime' variable

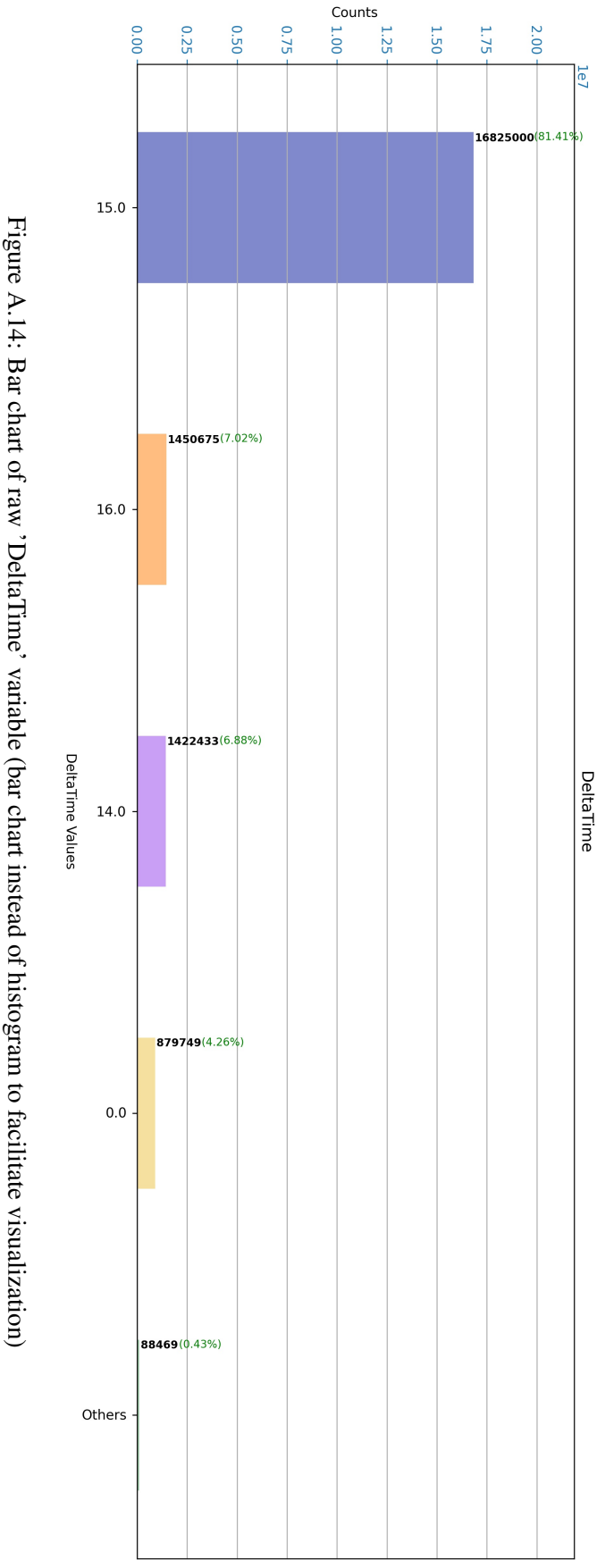


Figure A.14: Bar chart of raw 'DeltaTime' variable (bar chart instead of histogram to facilitate visualization)

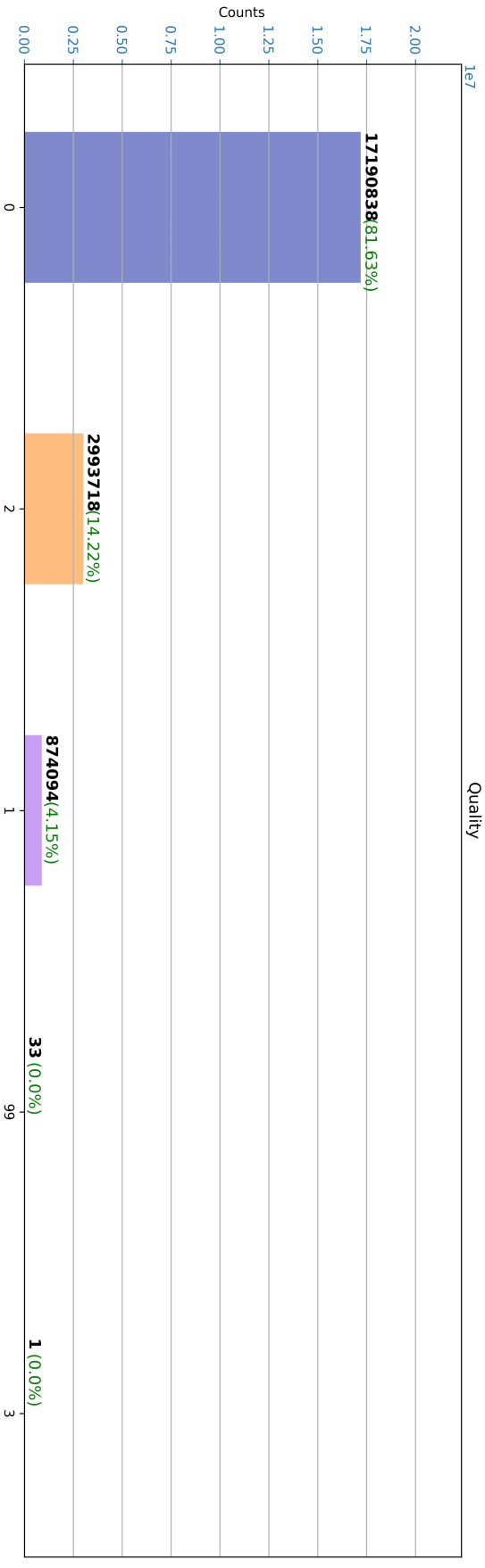


Figure A.15: Bar chart of raw 'Quality' variable

Appendix B

Data cleaning support

Beginning of Table B.1				
PUnit	PunitLine	Line	Real Line	Freq
Unit 01	1	0	1	223492
Unit 01	1	1	1	176760
Unit 01	1	16	1	9381
Unit 02	1	1	1	399089
Unit 03	7	3	3	371720
Unit 04	2	6	6	288356
Unit 05	2	5	5	140877
Unit 06	3	3	8	33849
Unit 06	3	8	8	327624
Unit 08	1	1	1	294325
Unit 09	3	3	3	372077
Unit 10	5	5	9	12323
Unit 10	5	9	9	292066
Unit 11	7	7	7	391391
Unit 12	4	4	5	26259
Unit 12	4	5	5	345543
Unit 13	1	1	1	411892
Unit 14	10	10	10	415091
Unit 15	7	9	9	382543
Unit 16	3	5	5	399991
Unit 17	6	6	6	443905
Unit 18	8	3	3	402080
Unit 19	8	4	4	418332
Unit 21	2	8	8	70440
Unit 22	7	6	6	433378
Unit 23	2	2	2	389825
Unit 24	5	5	5	421095
Unit 25	9	7	7	149660
Unit 25	9	9	7	243626
Unit 26	4	4	4	89285
Unit 27	6	9	9	375205

Continuation of Table B.1				
PUnit	PunitLine	Line	Reality	Nr. records
Unit 29	7	7	7	340821
Unit 30	6	8	8	75982
Unit 31	10	10	10	414386
Unit 33	6	6	6	380624
Unit 34	7	9	9	422684
Unit 35	6	6	6	376992
Unit 36	3	8	8	389095
Unit 37	2	2	2	428825
Unit 38	4	4	4	369998
Unit 39	8	8	9	269017
Unit 39	8	9	9	37887
Unit 40	8	8	8	408631
Unit 41	9	2	2	421212
Unit 42	5	5	5	420737
Unit 43	3	10	10	134674
Unit 43	3	3	10	169370
Unit 44	5	3	3	312454
Unit 44	5	5	3	110032
Unit 45	1	1	1	399426
Unit 46	6	7	7	303887
Unit 47	10	10	10	337288
Unit 48	5	5	5	384534
Unit 49	2	2	2	432961
Unit 50	8	3	3	346581
Unit 51	9	9	9	363422
Unit 53	9	2	2	56656
Unit 53	9	9	2	345025
Unit 54	10	10	10	355704
Unit 55	10	10	10	344094
Unit 57	8	8	8	316670
Unit 58	9	7	7	107957
Unit 58	9	9	7	239009
Unit 59	5	4	4	380580
Unit 60	10	3	3	351651
Unit 61	4	4	4	347408
Unit 62	3	3	6	242073
Unit 62	3	6	6	166806
Error rate:	47,85%	9,59%	Total corks:	20056633

Table B.1: Correspondence of units to lines according to variables 'PunitLine' and 'Line'

Appendix C

Tables, bar charts and histogramms of the clean data variables

Variable	Oldest	Youngest	Max Difference
AnalTime	07/09/2020 23:59	27/11/2020 14:57	0 days 06:21:23

Table C.1: Summary of variable clean 'AnalysisDate' (previously called 'AnalTime')

Variable	Nr. Classes	Modal Class	Mode Count	Min. Class	Min. Class Count
Line	10	6	2 328 218	4	1 603 881
Module	56	Unit 17	443 572	Unit 30	75 886
Chamber	8	4	2 527 568	7	2 418 104
Shift	2	Turno 1	10 195 255	Turno 2	9 597 590
Lot	208	PS 25839	908 847	0000646110	4
Class	28	EXT	7 558 849	FLOR ESP (W E) R15	12 115
Calibre	8	49x24	8 120 050	52x24	37 704
Quality	3	0	16 089 289	1	820 681
Origin	8	AC	11 338 895	L	7 668
Error	8	18680549	0	5	3

Table C.2: Summary of the clean categorical variables (?module' was previously called 'PUnit')

Variable	Mean	Std. Dev.	Median	Min.	Q1	Q3	Max.	Skewness	Kurtosis
Module	7.330	68.550	2.503	0.000	0.248	4.849	43622.910	104.584	36945.105
PeakTime	3.814	1.163	3.600	0.000	3.100	4.500	11.400	0.700	1.514
AnalysisDuration	23553.900	9471209.843	15.000	0.000	15.000	15.000	3813856109.000	402.000	161.985

Table C.3: Summary of the clean numerical variables ('PeakValue' is measured in mV.s; 'PeakTime' and 'AnalysisDuration' are in seconds; 'AnalysisDuration was previously called 'DeltaTime')

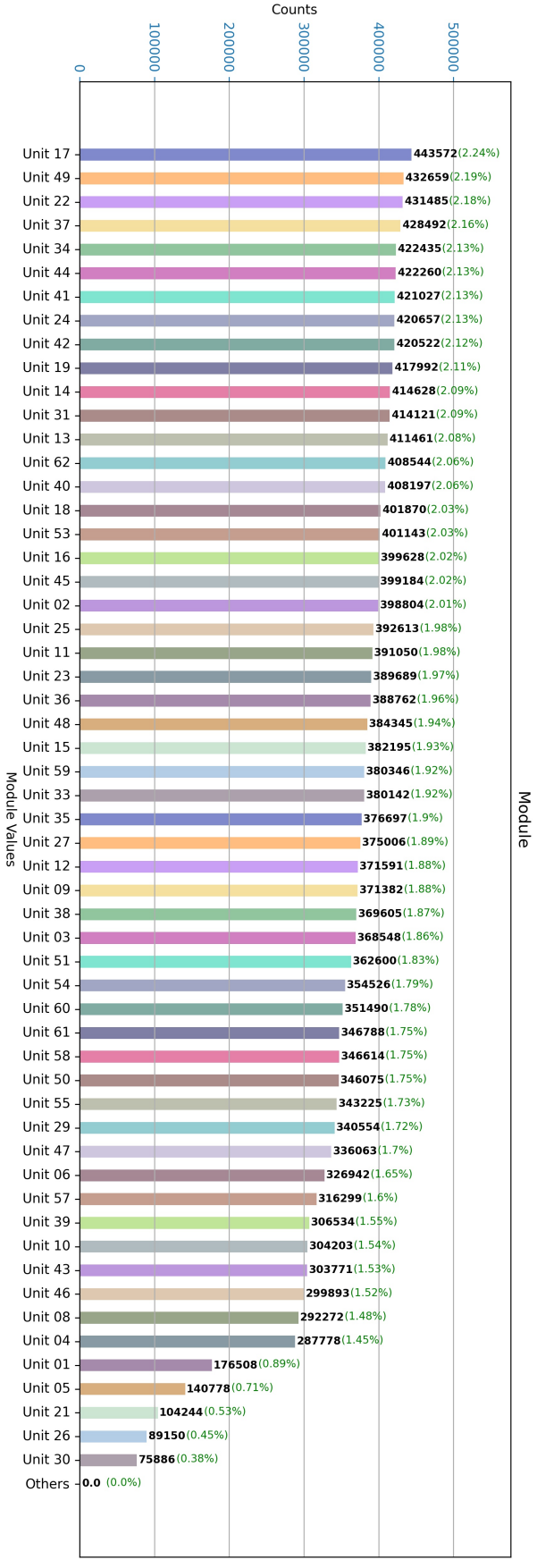


Figure C.1: Bar chart of clean 'Module' variable (previously called 'PUnit')

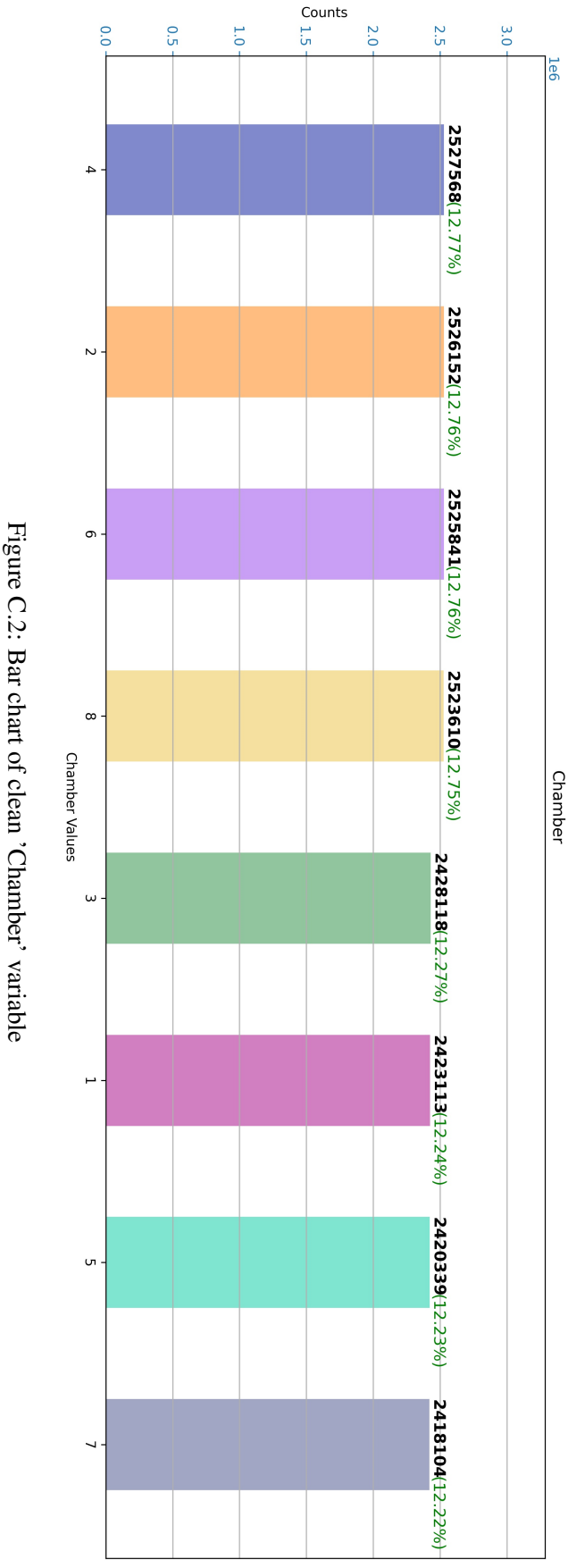


Figure C.2: Bar chart of clean 'Chamber' variable

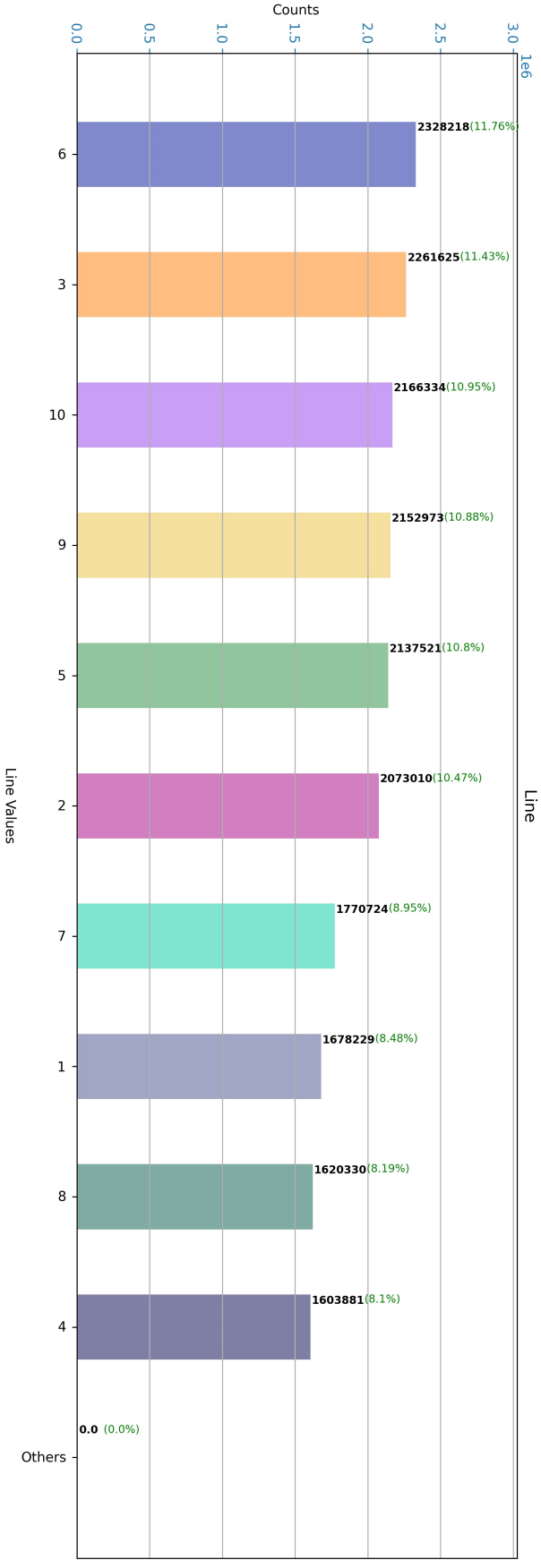


Figure C.3: Bar chart of clean 'Line' variable

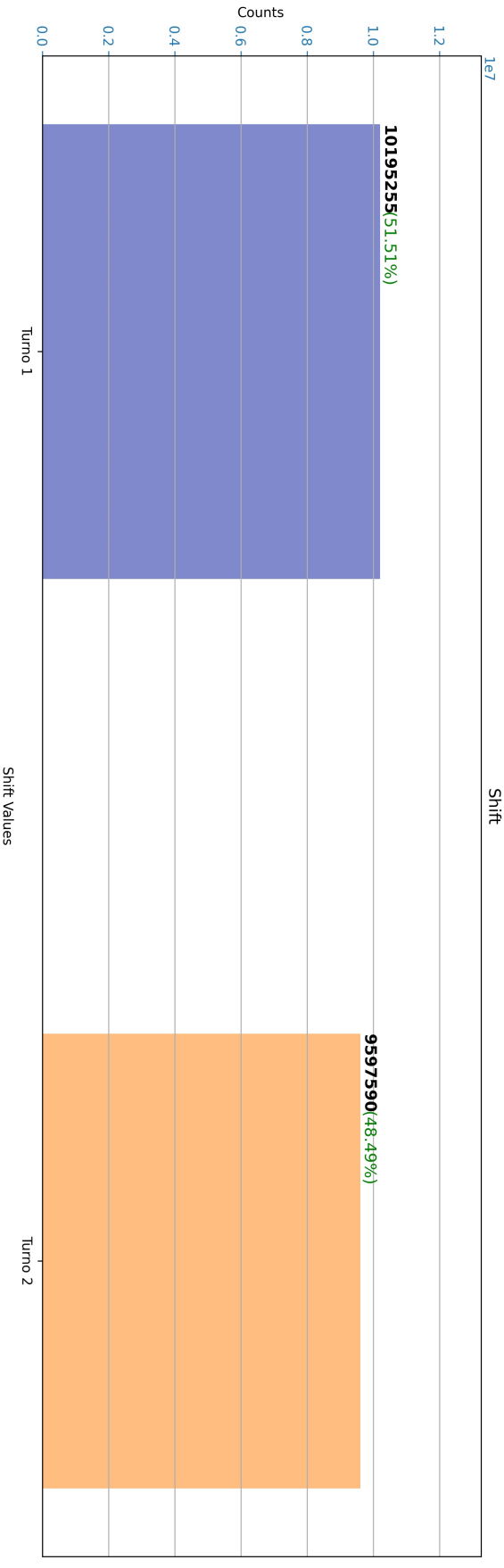


Figure C.4: Bar chart of clean 'Shift' variable

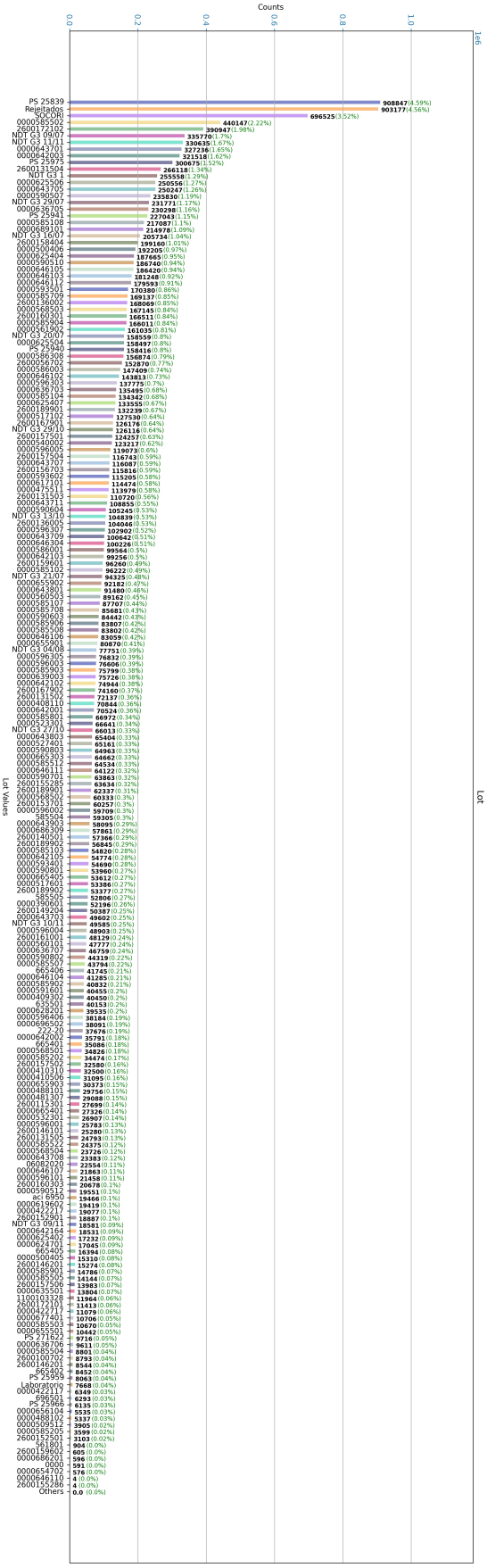


Figure C.5: Bar chart of clean 'Lot' variable

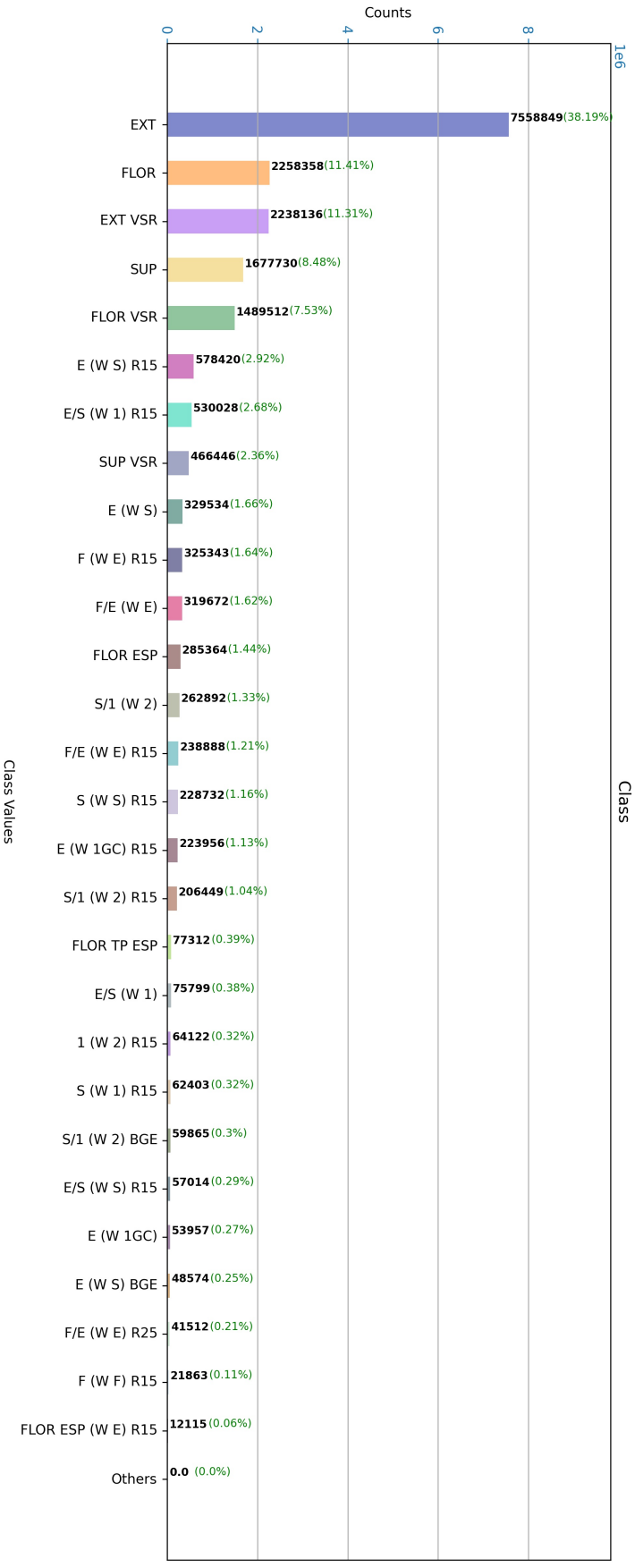


Figure C.6: Bar chart of clean 'Class' variable

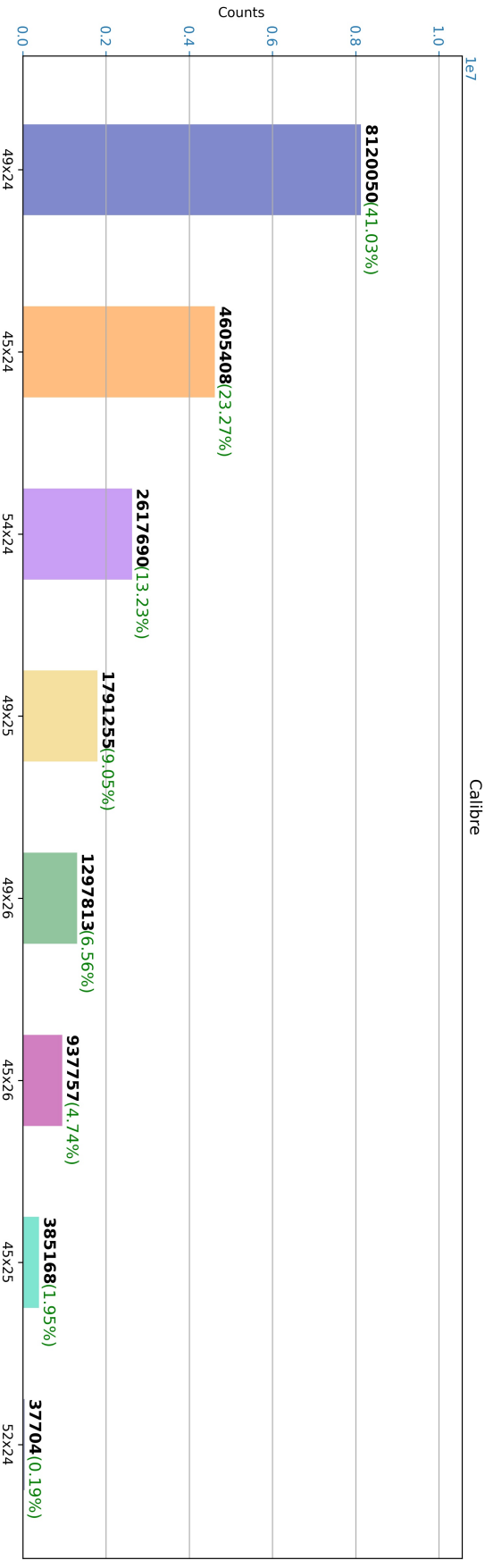


Figure C.7: Bar chart of clean 'Calibre' variable

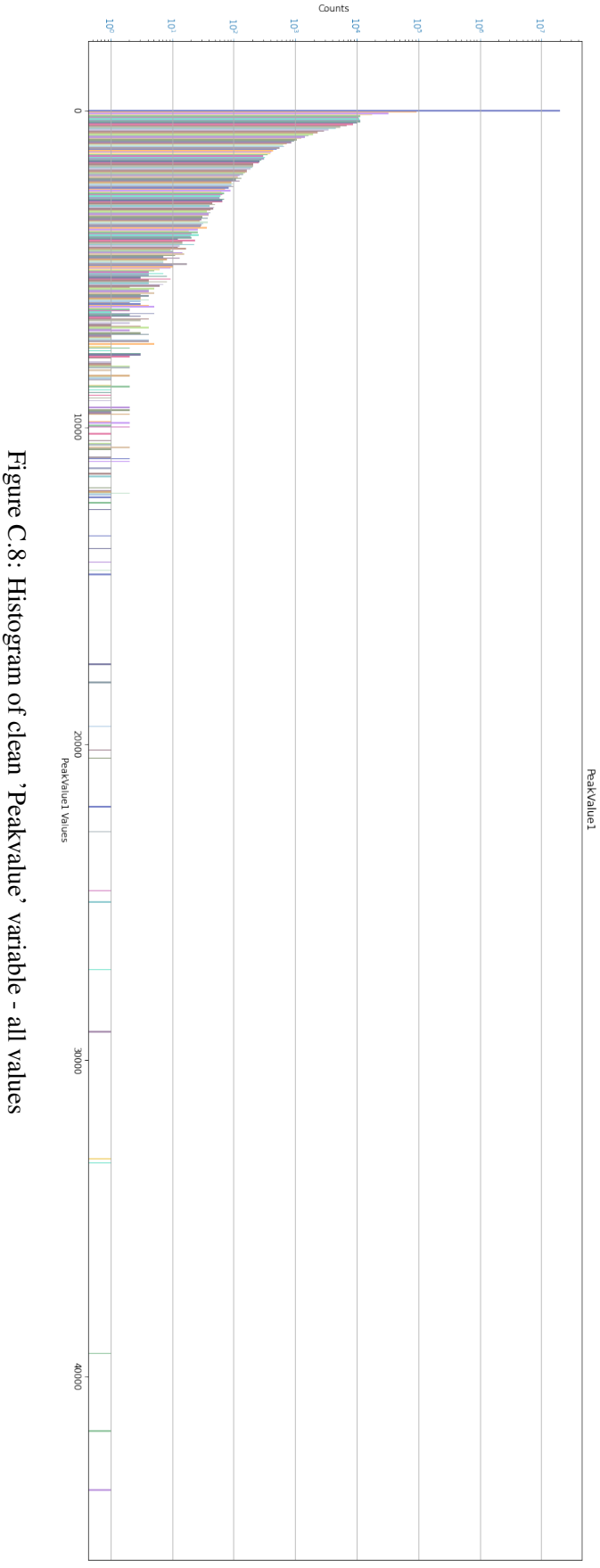


Figure C.8: Histogram of clean 'Peakvalue' variable - all values

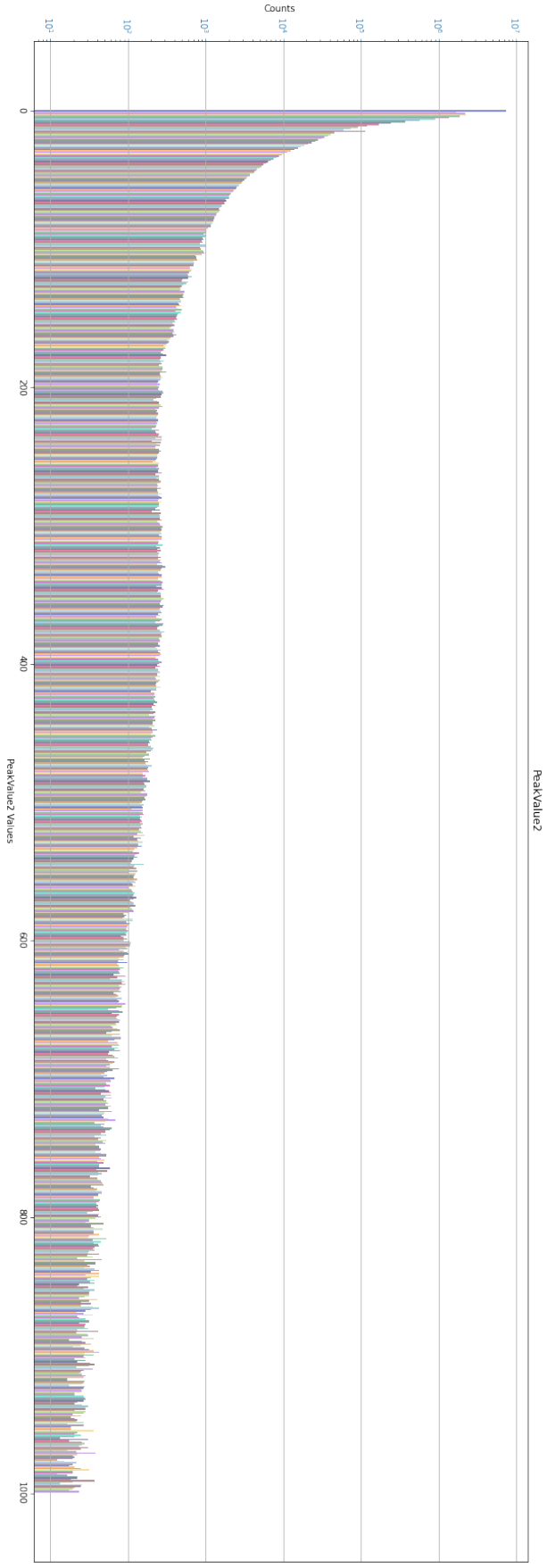


Figure C.9: Histogram of clean 'Peakvalue' variable - values up to 1000 mV.s

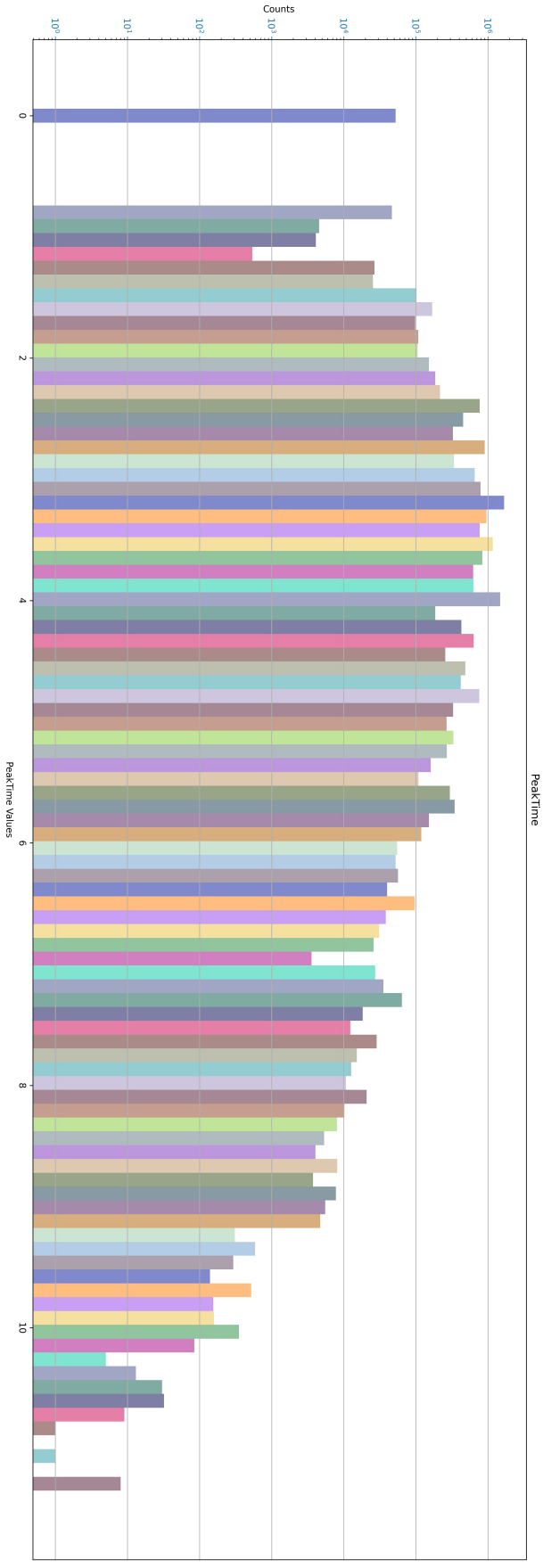


Figure C.10: Histogram of clean 'PeakTime' variable

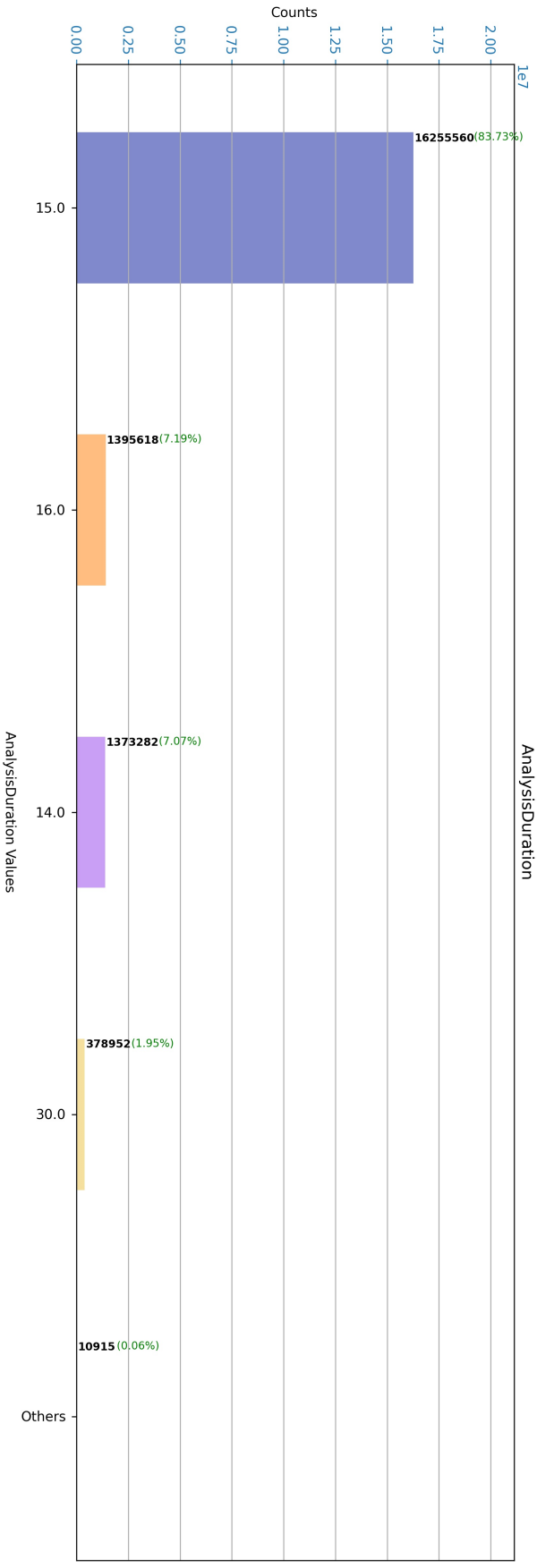


Figure C.11: Bar chart of clean 'AnalysisDuration' variable (previously called 'DeltaTime'; bar chart instead of histogram to facilitate visualization)

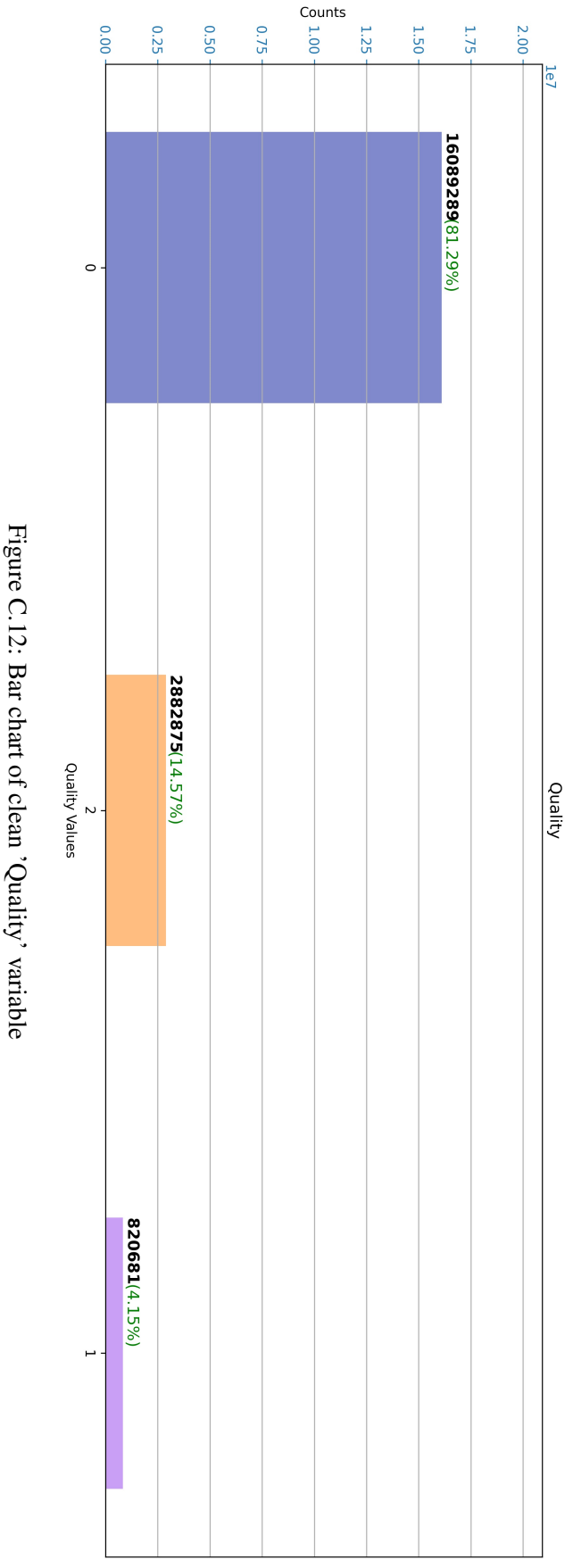


Figure C.12: Bar chart of clean 'Quality' variable

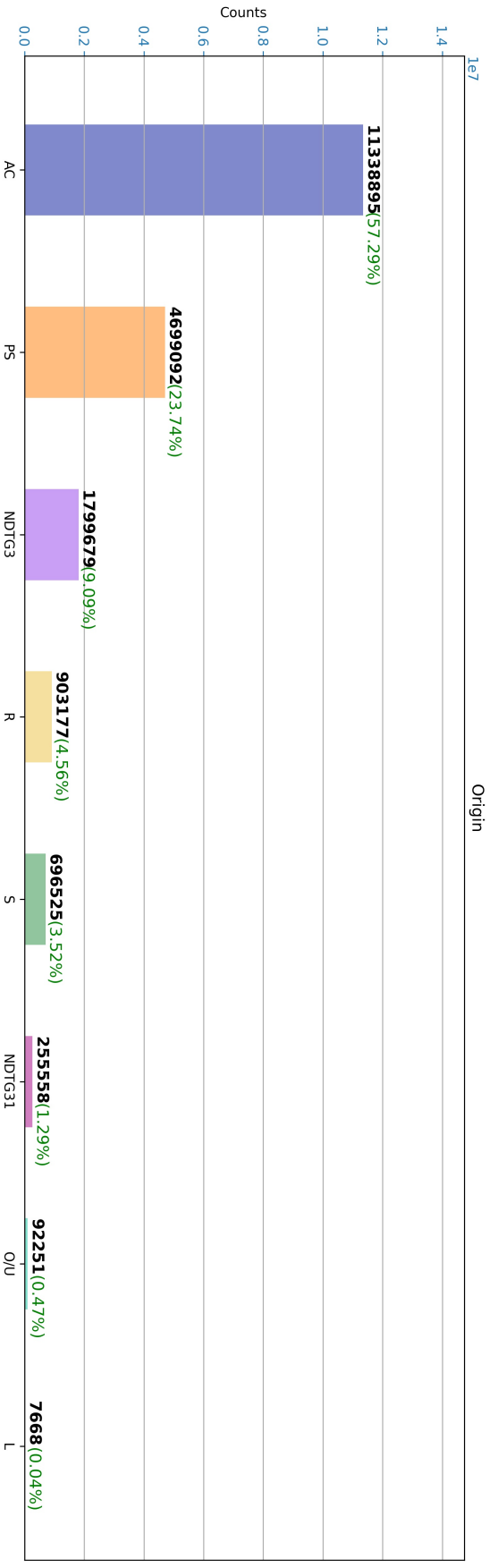


Figure C.13: Bar chart of the created 'Origin' variable

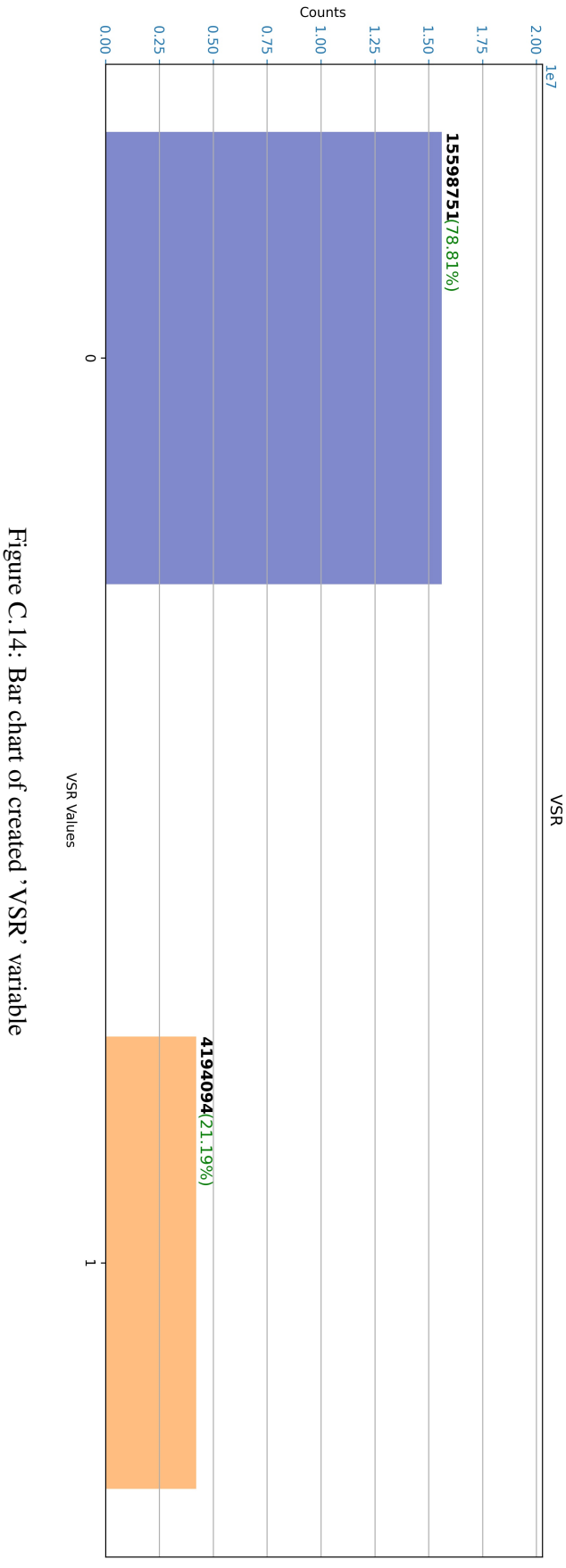


Figure C.14: Bar chart of created 'VSR' variable



Figure C.15: Bar chart of created 'Error' variable

Appendix D

Relationships between variables

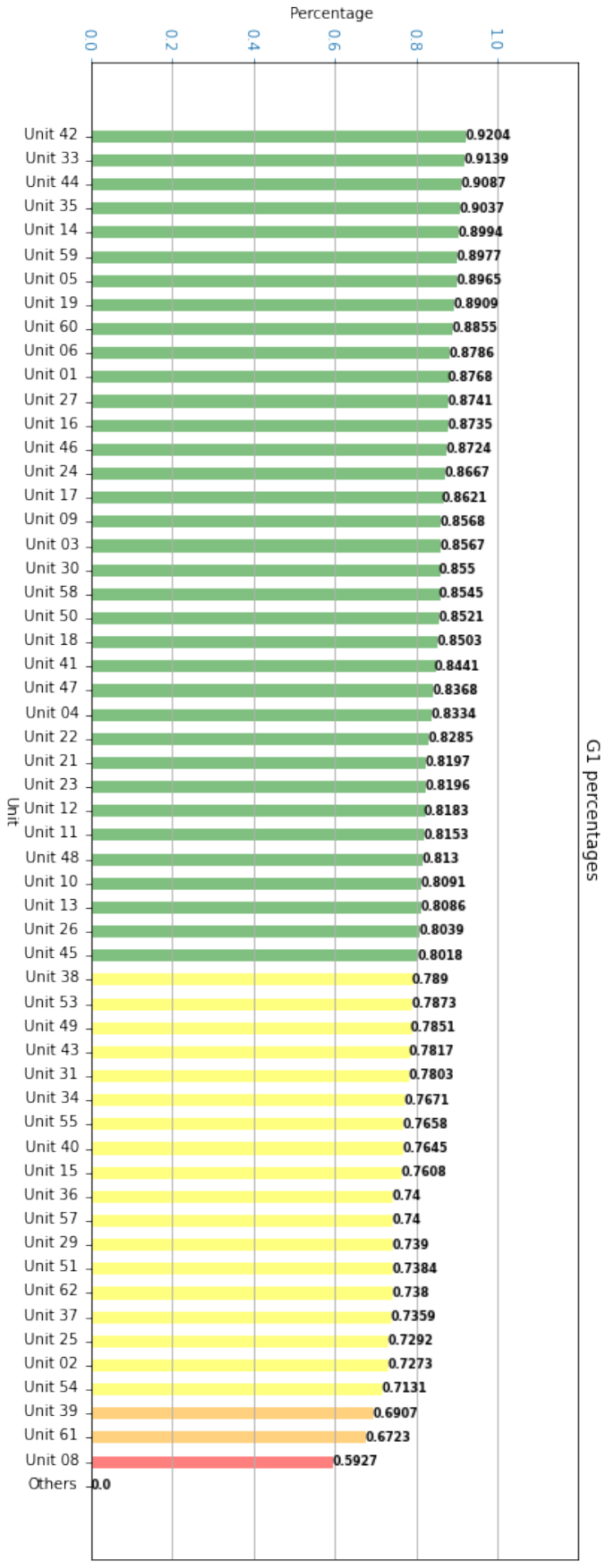


Figure D.1: Percentage of classified G1 stoppers per Module

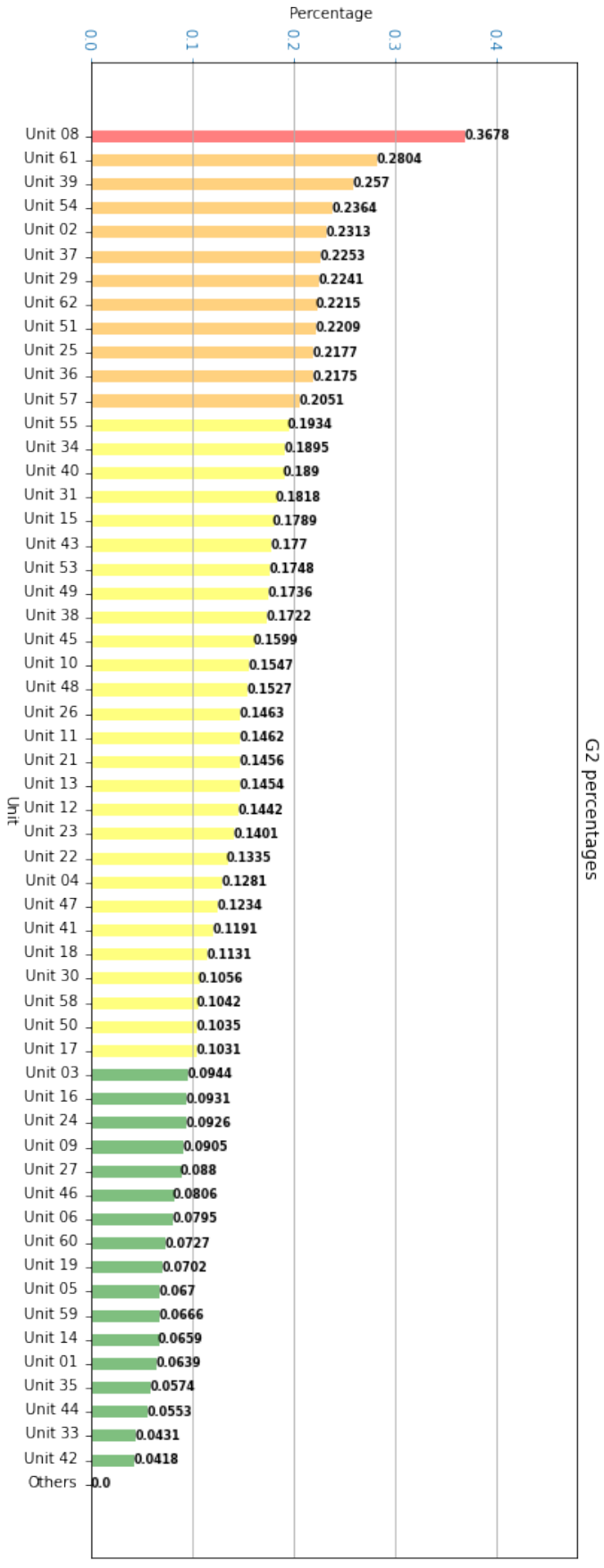


Figure D.2: Percentage of classified G2 stoppers per Module

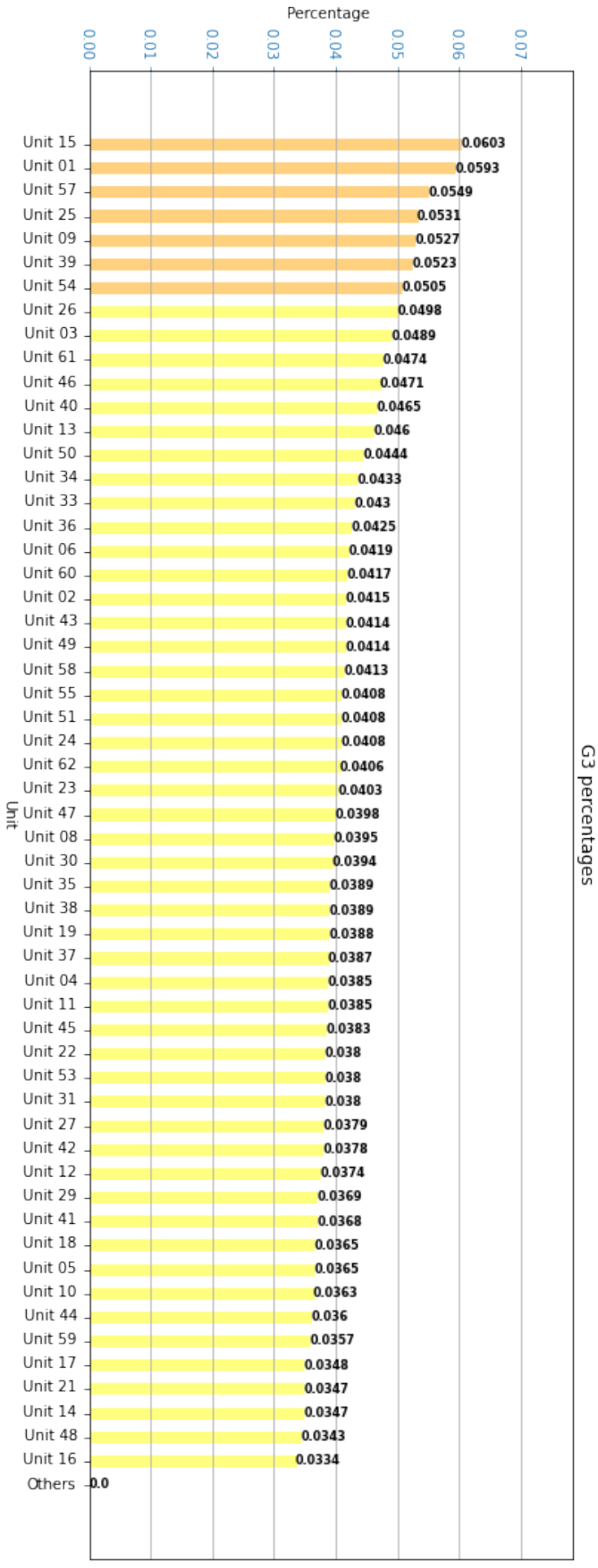


Figure D.3: Percentage of classified G3 stoppers per Module

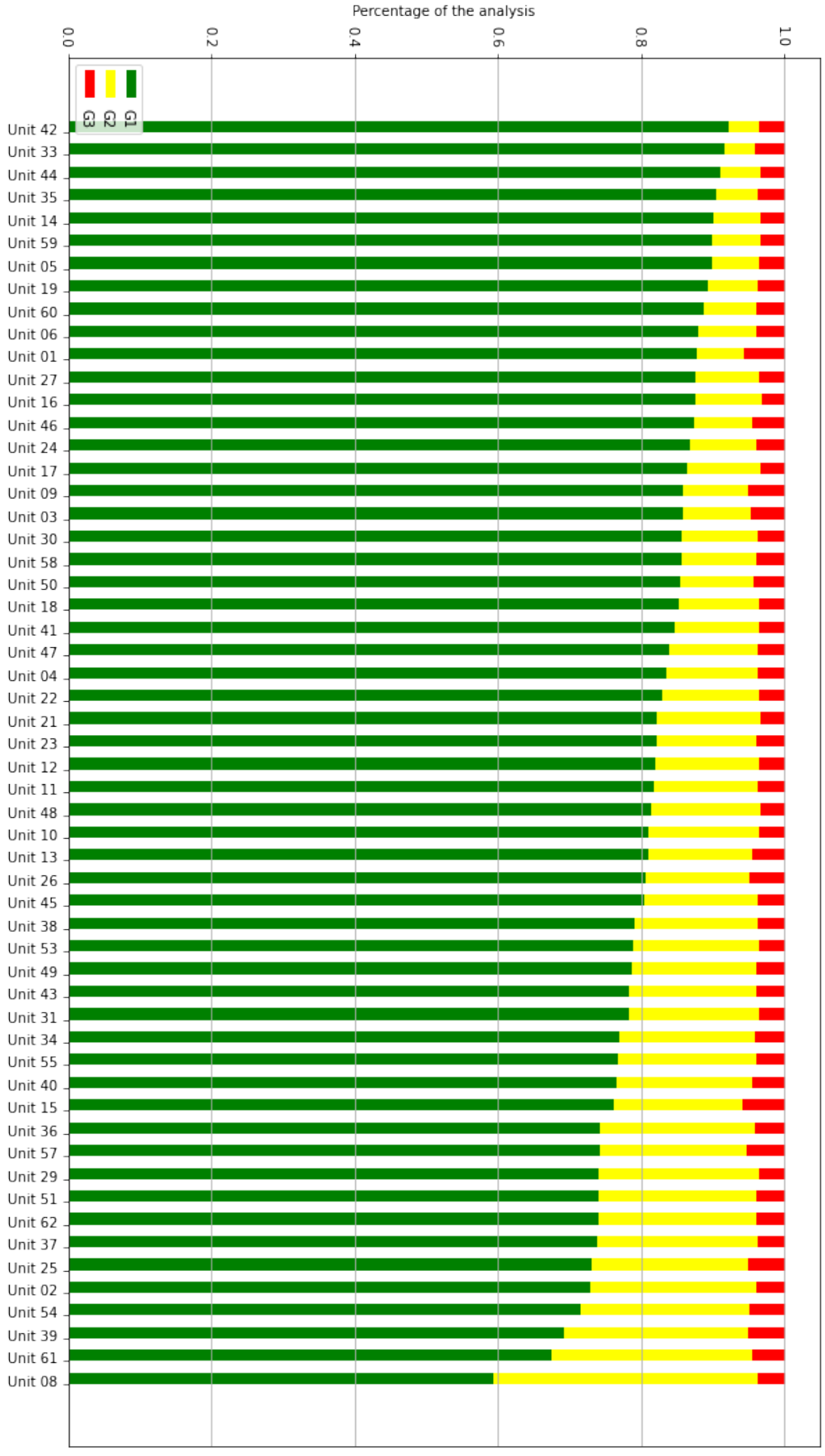


Figure D.4: Stacked percentages of stoppers classified for each group per Module

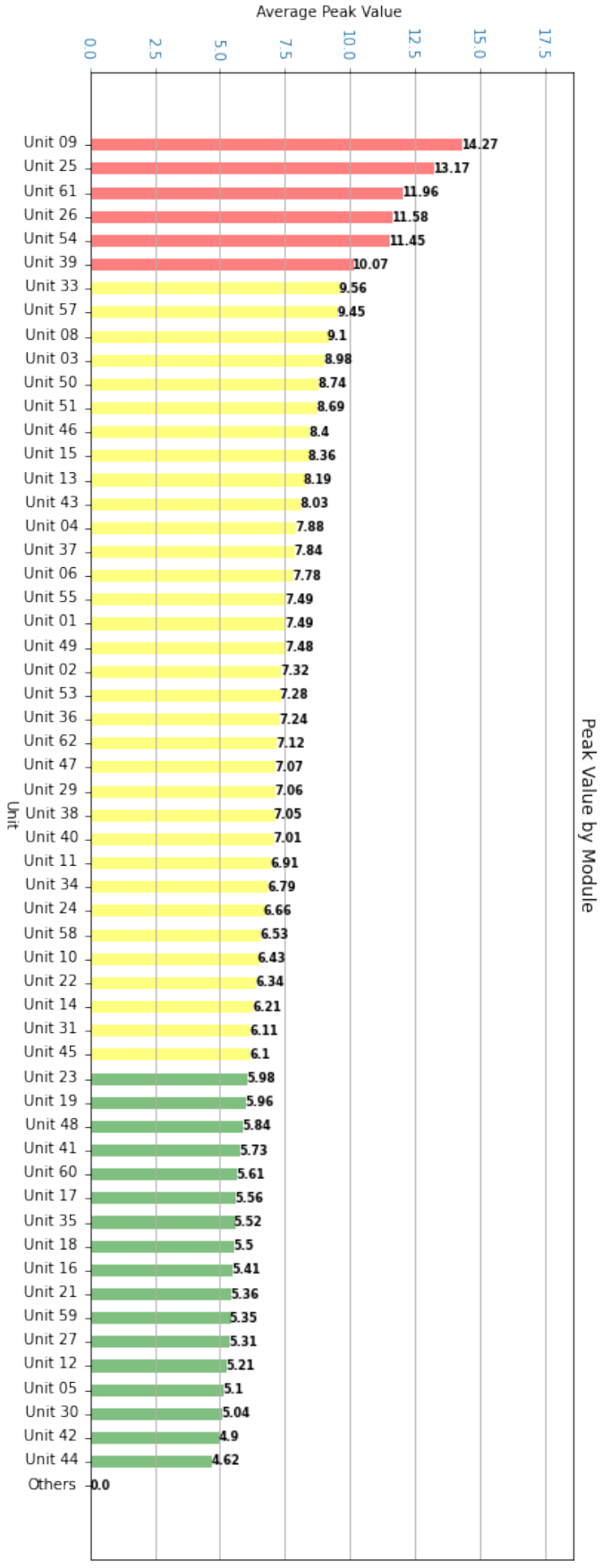


Figure D.5: Average PeakValue per Module

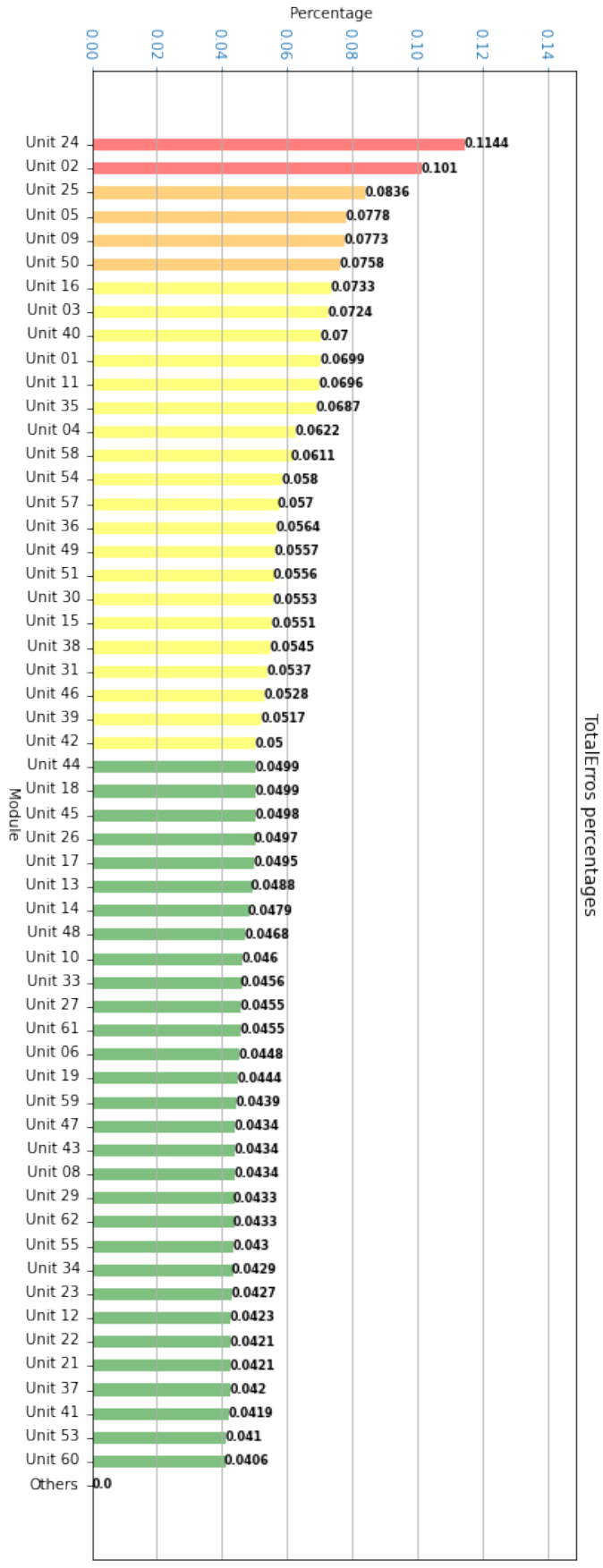


Figure D.6: Percentage of observations with error per Module

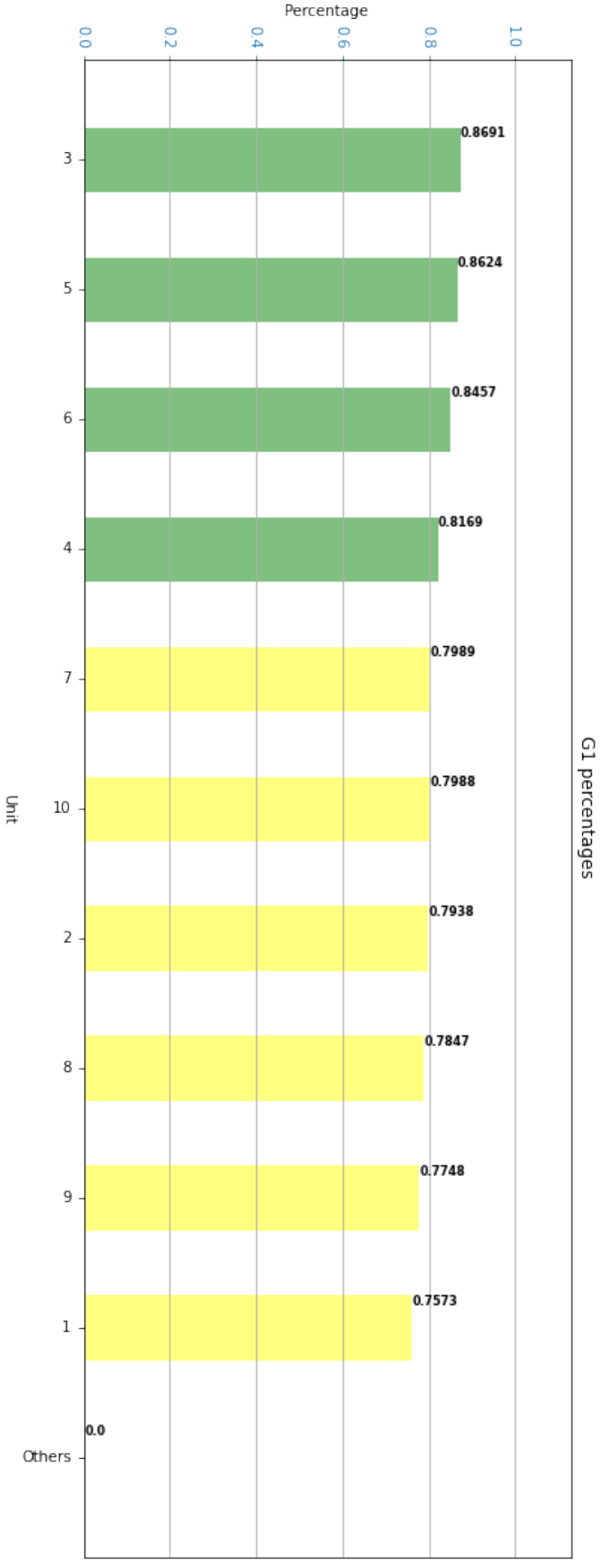


Figure D.7: Percentage of classified G1 stoppers per Line

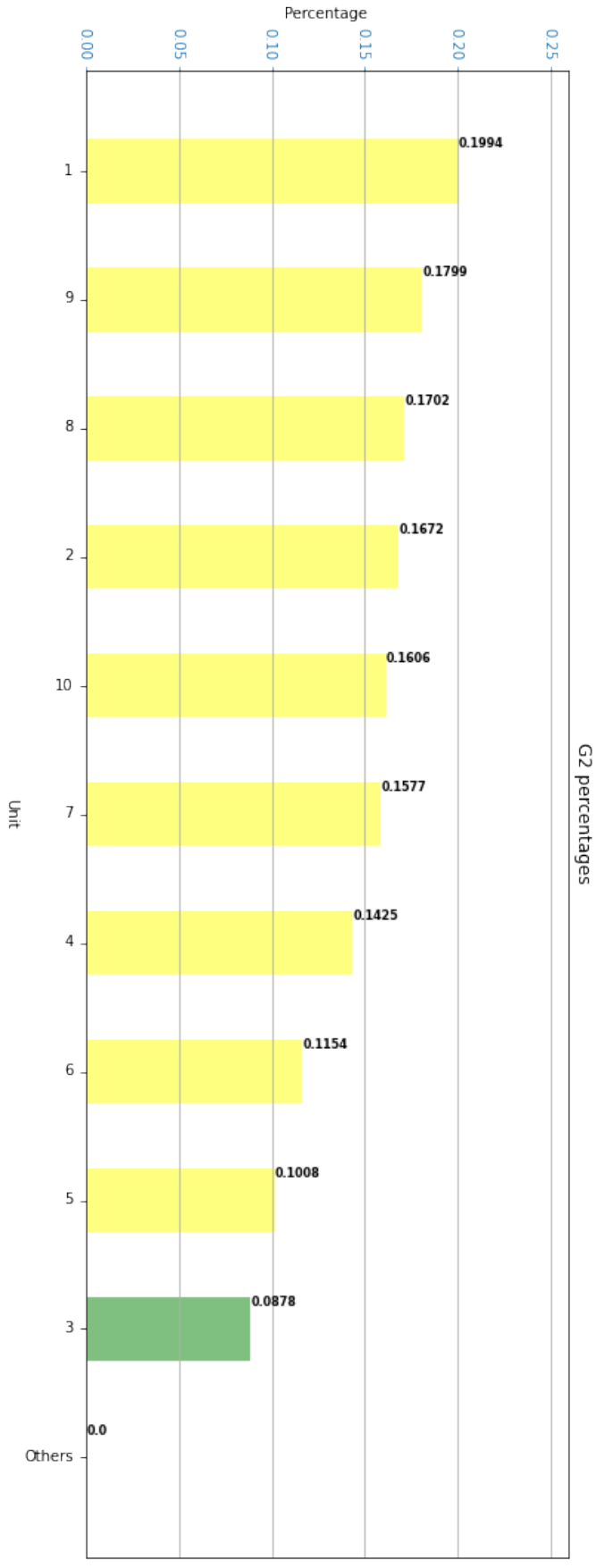


Figure D.8: Percentage of classified G2 stoppers per Line

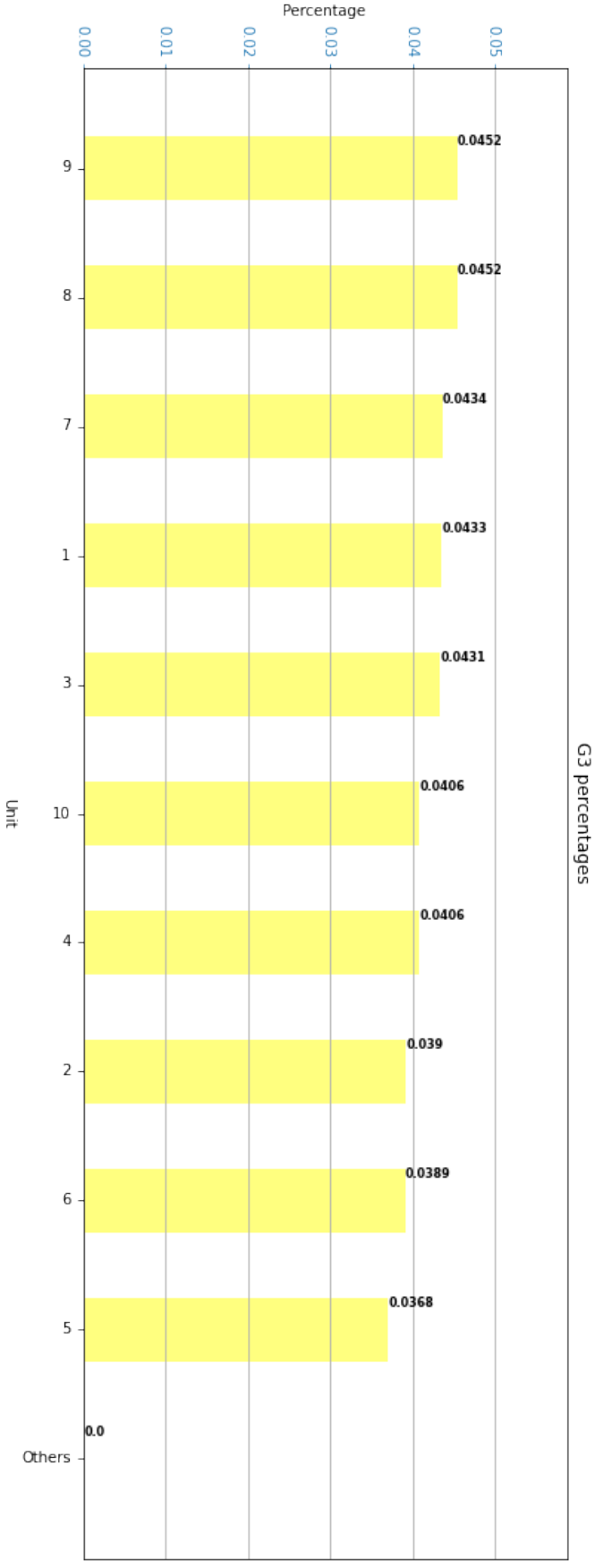


Figure D.9: Percentage of classified G3 stoppers per Line

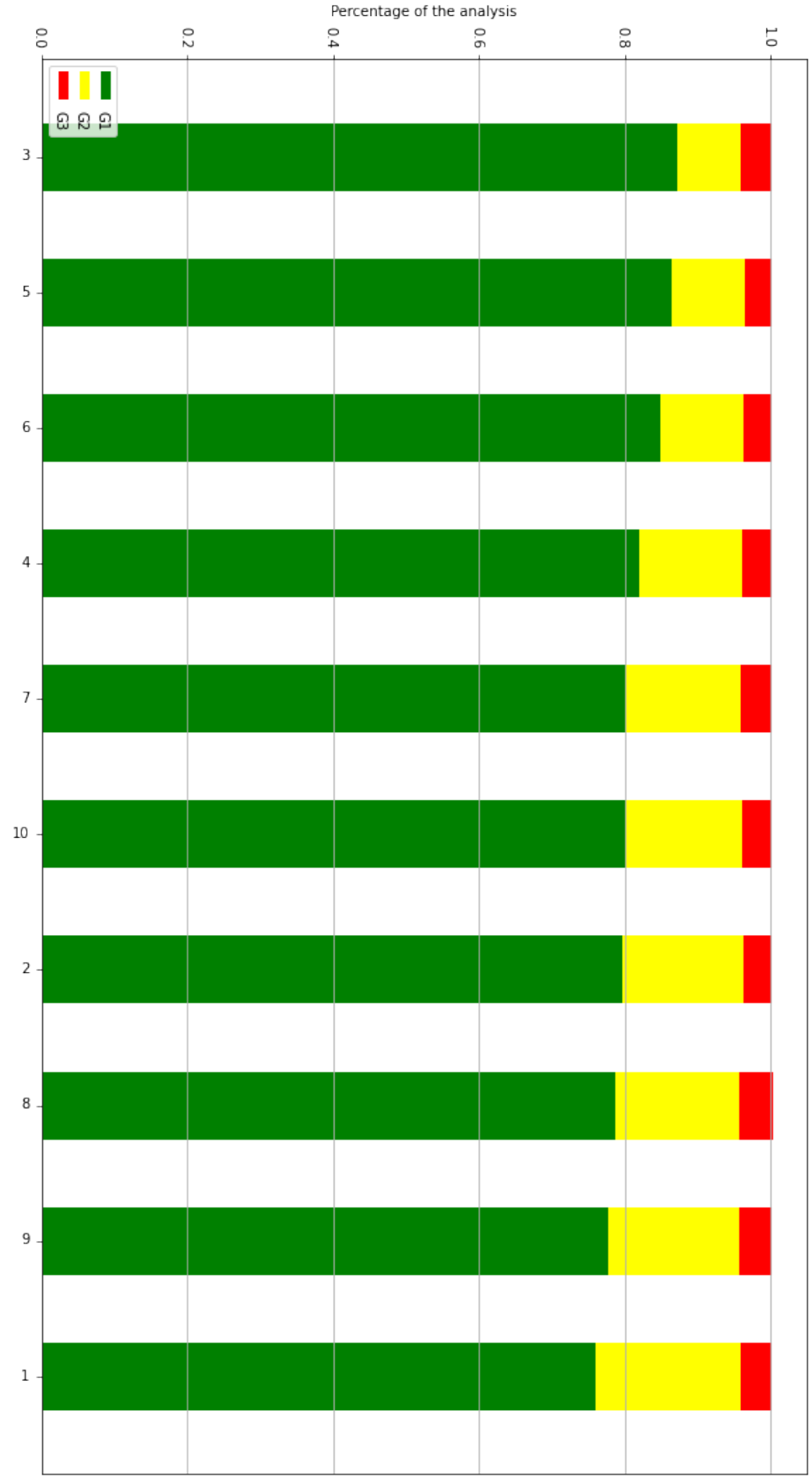


Figure D.10: Stacked percentages of stoppers classified for each group per Line

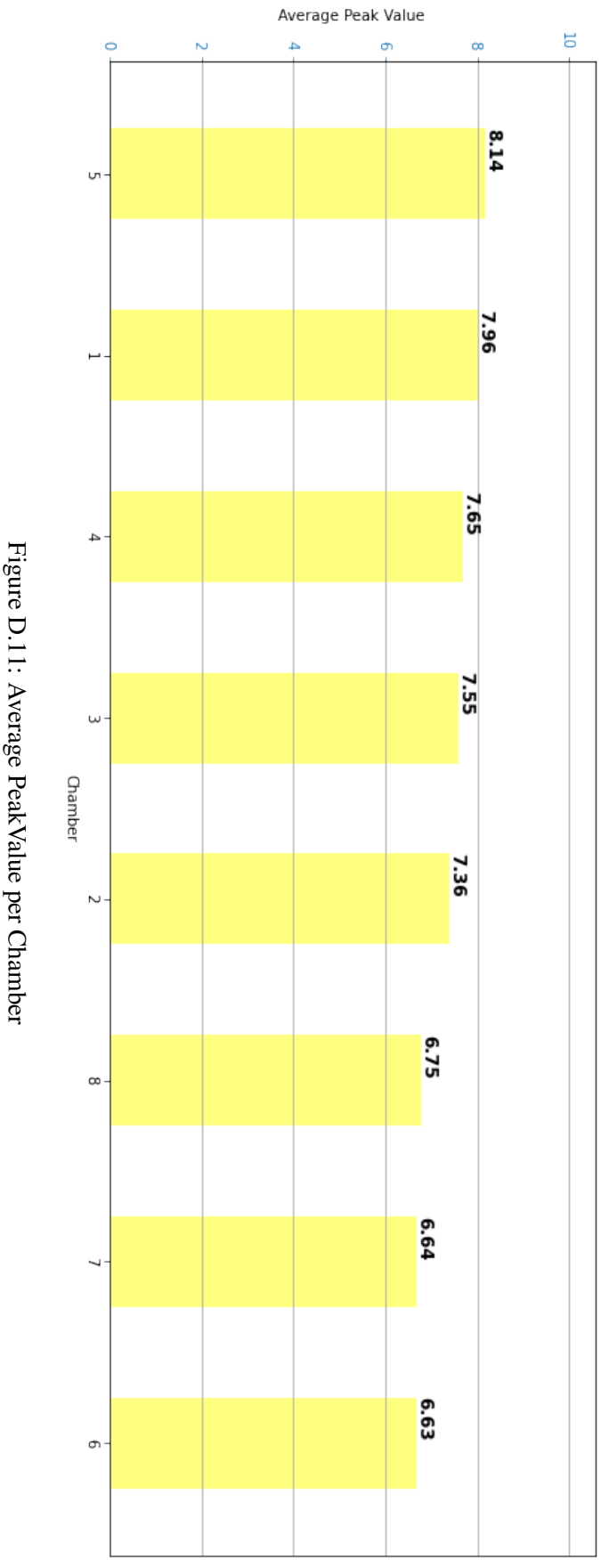


Figure D.11: Average PeakValue per Chamber

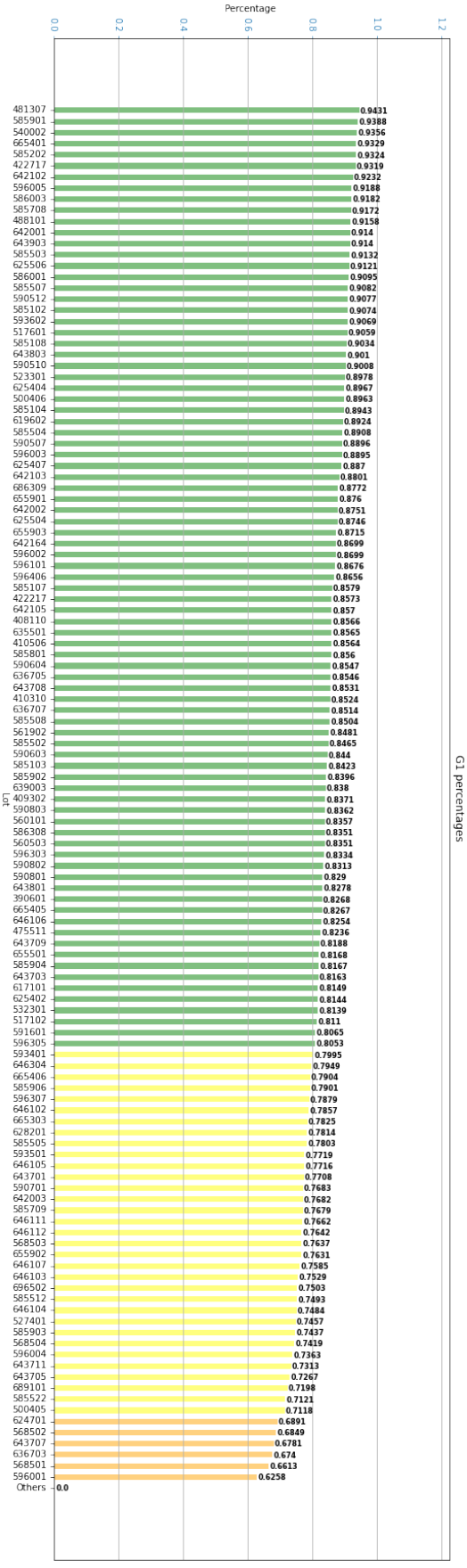


Figure D.12: Percentage of classified G1 stoppers per Lot

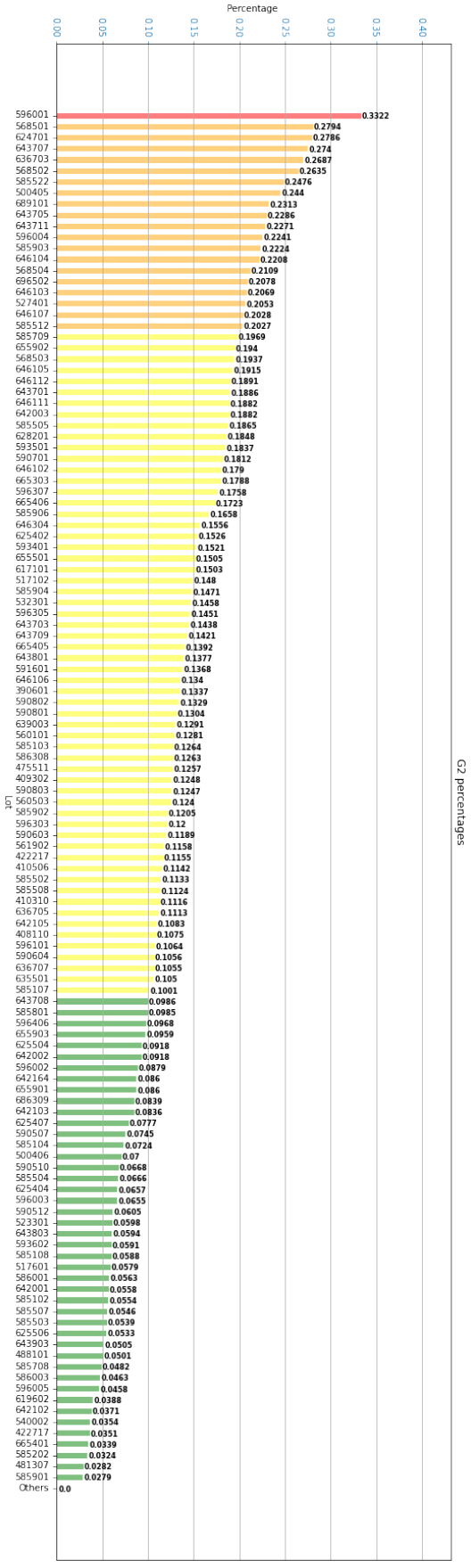


Figure D.13: Percentage of classified G2 stoppers per Lot

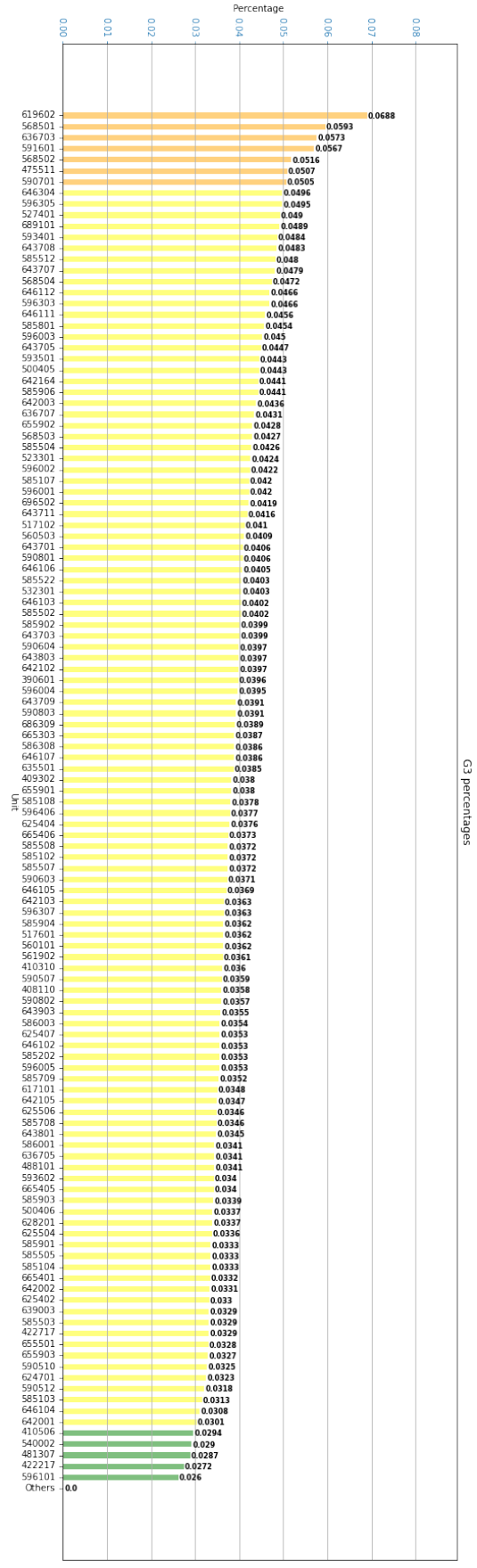


Figure D.14: Percentage of classified G3 stoppers per Lot

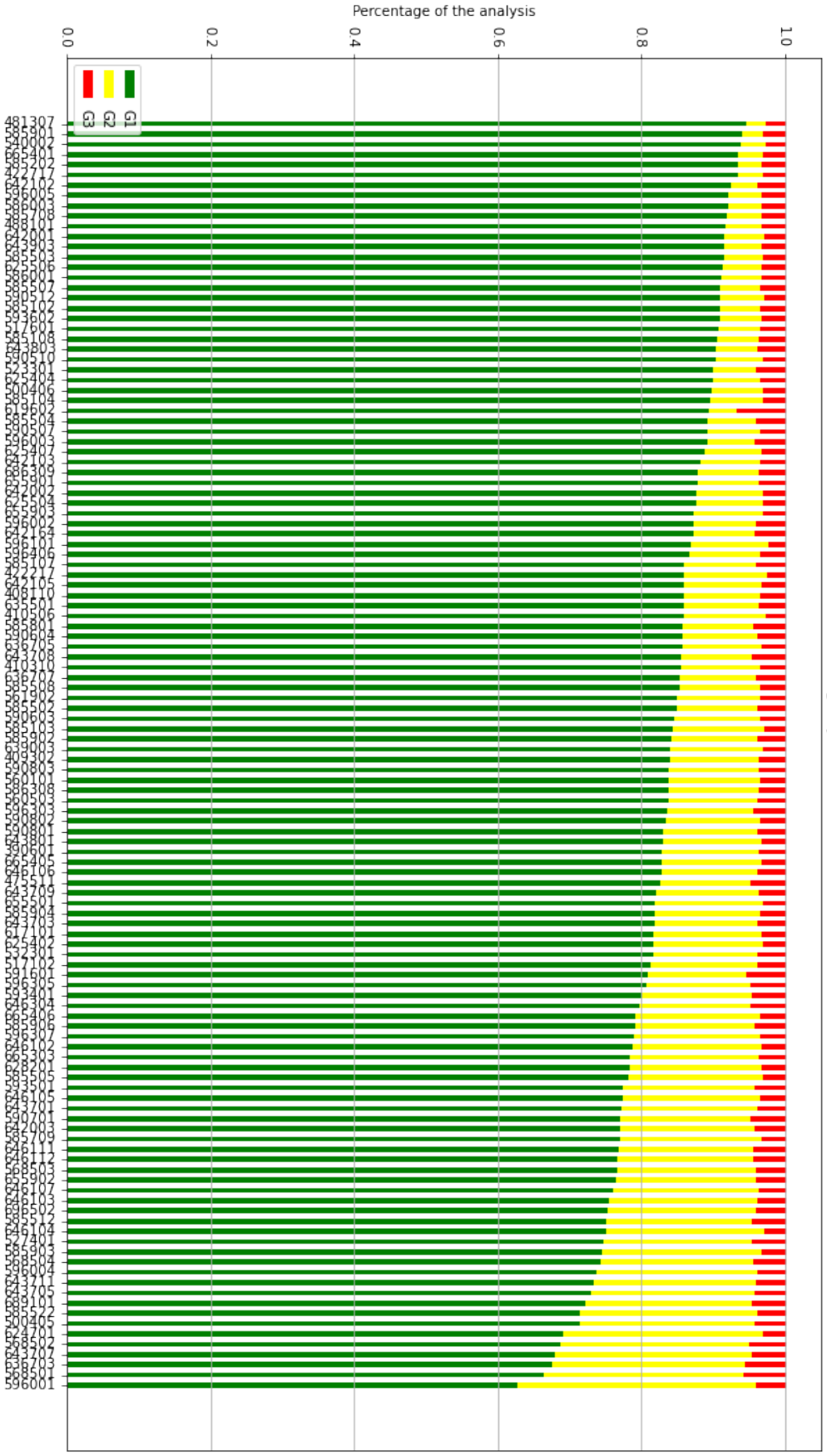


Figure D.15: Stacked percentages of stoppers classified for each group per Lot

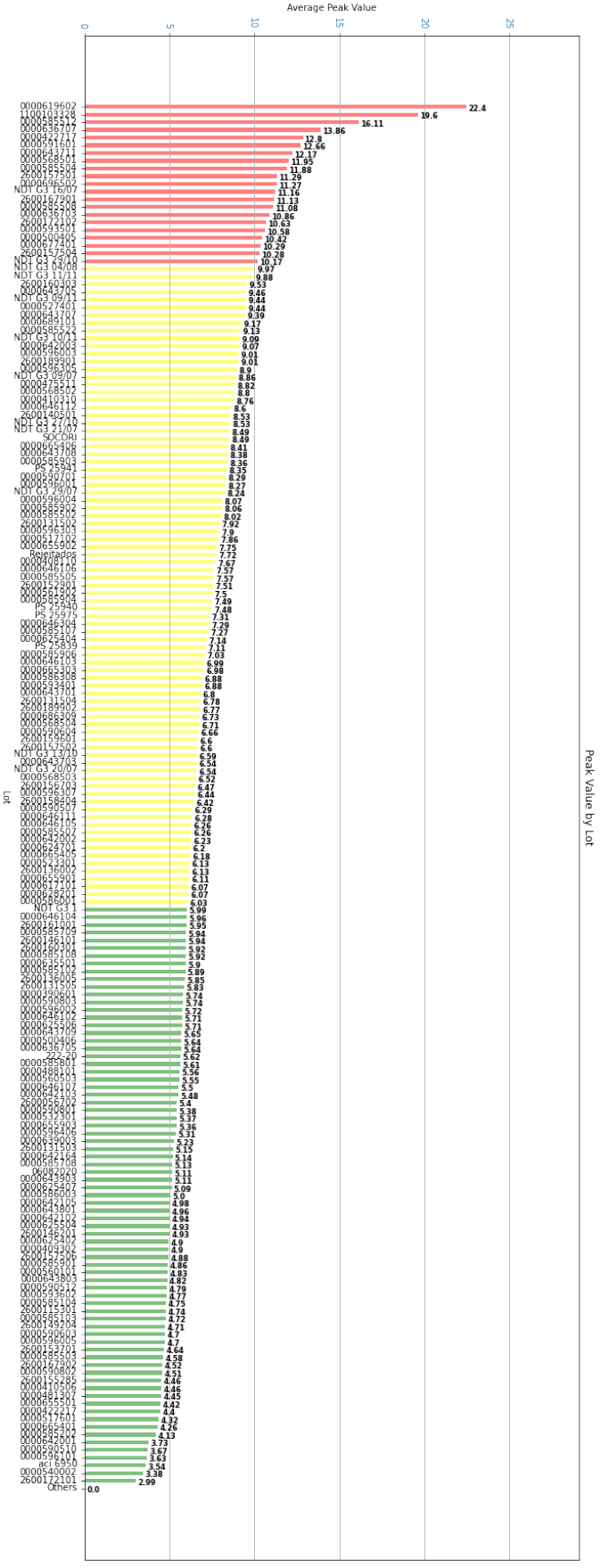


Figure D.16: Average PeakValue per Lot

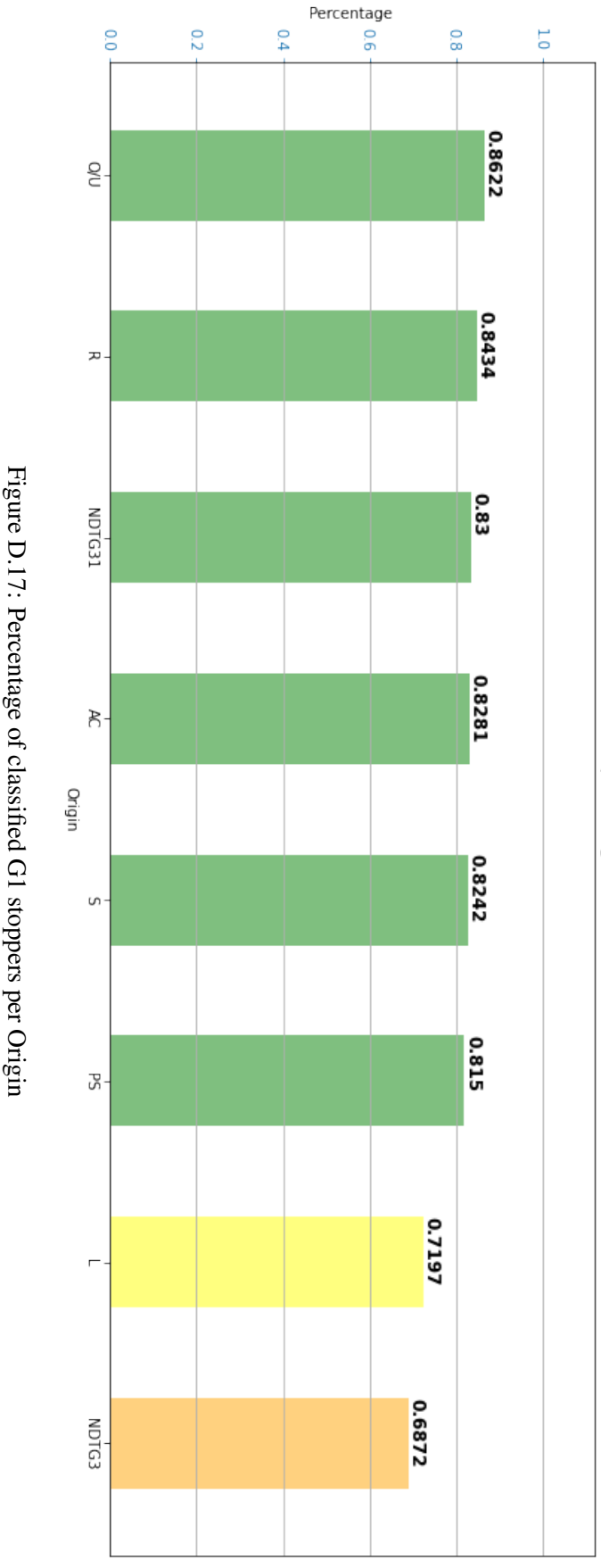


Figure D.17: Percentage of classified G1 stoppers per Origin

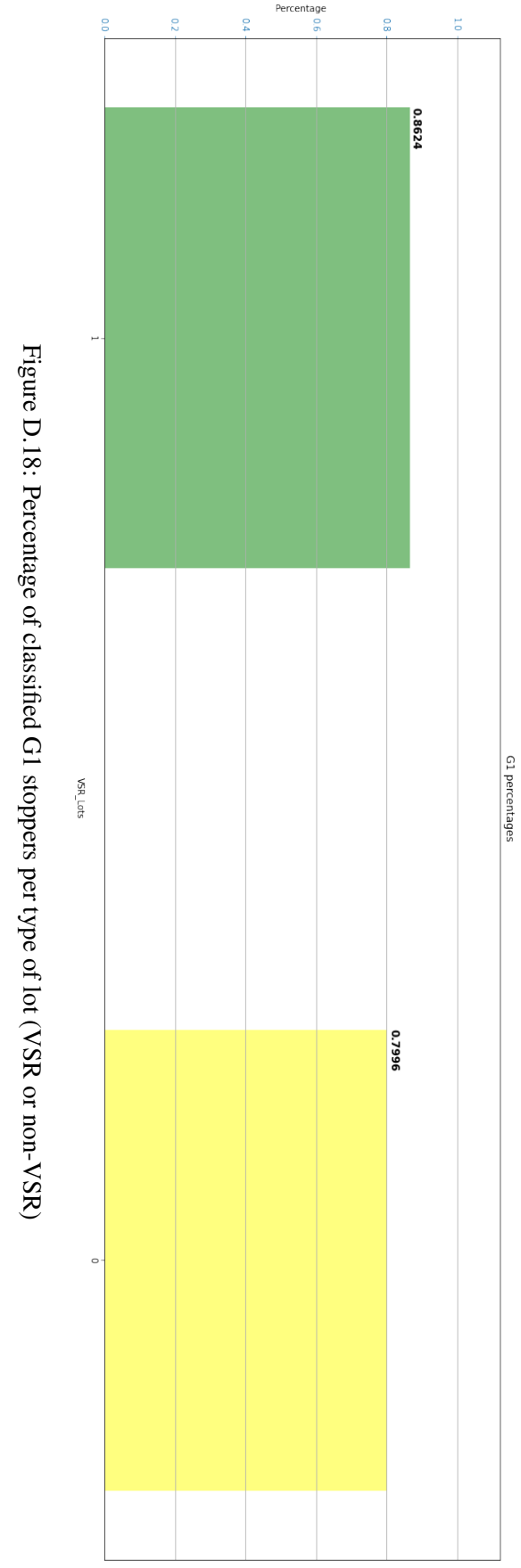


Figure D.18: Percentage of classified G1 stoppers per type of lot (VSR or non-VSR)

Appendix E

Anomalous patterns obtained with the clustering approach

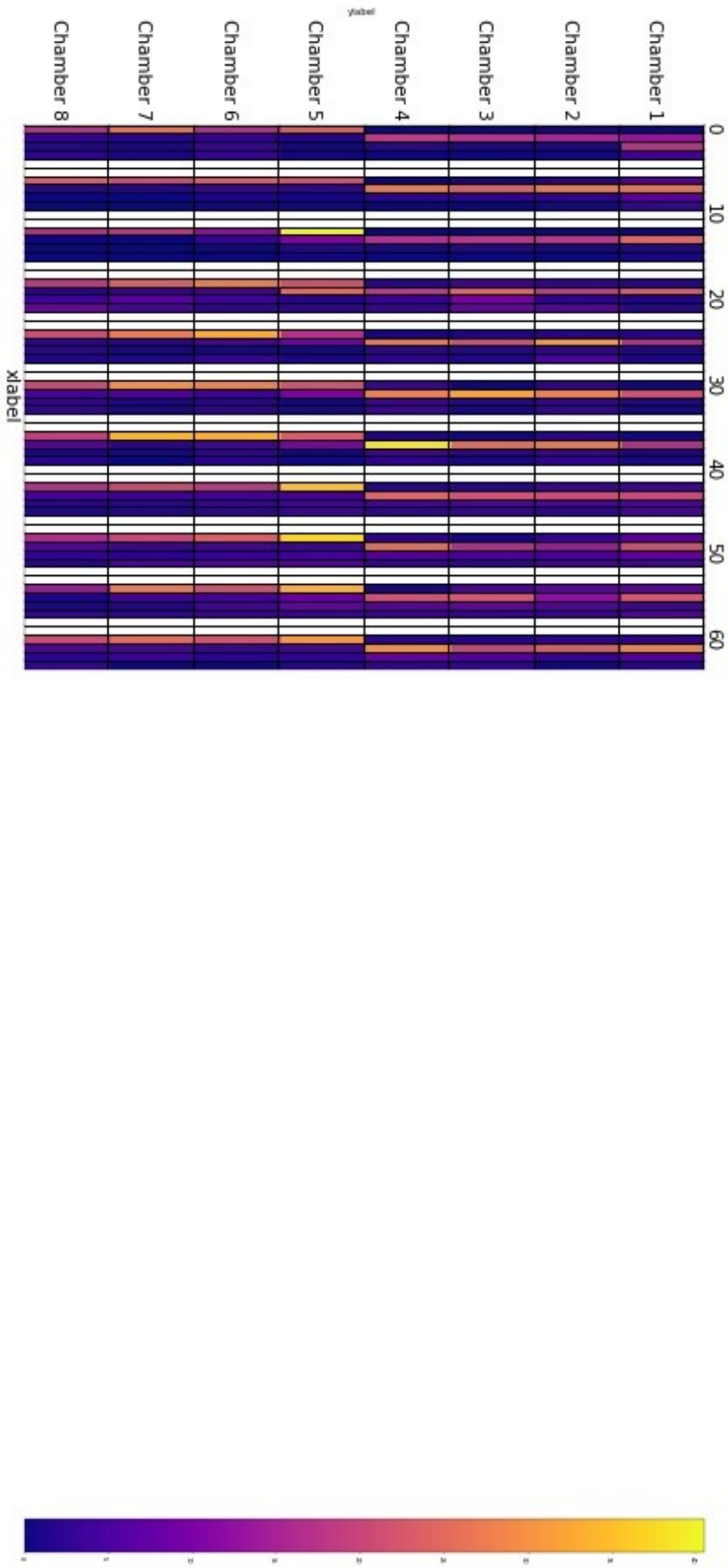


Figure E.1: 8G2 pattern

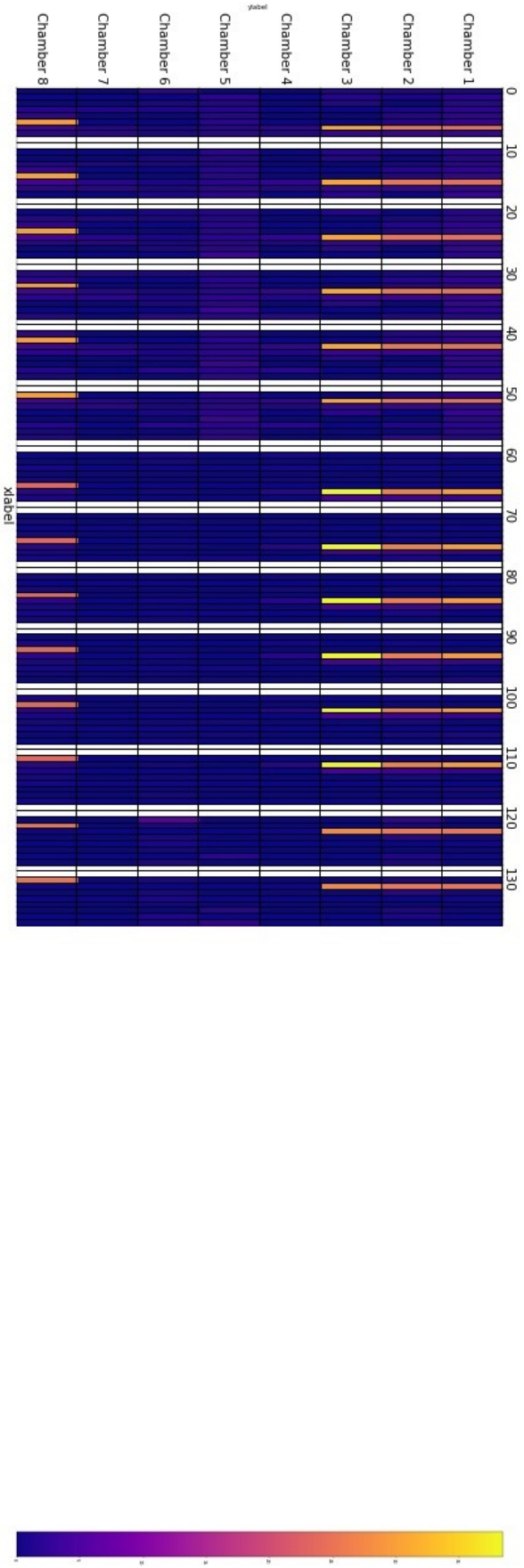


Figure E.2: 4G2 pattern

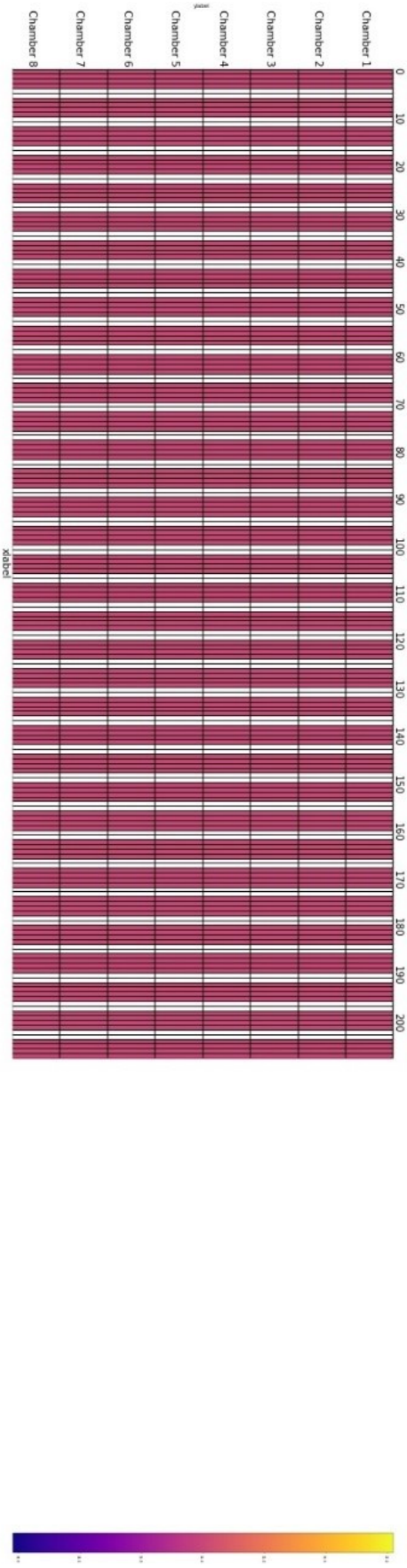


Figure E.3: Integration error pattern

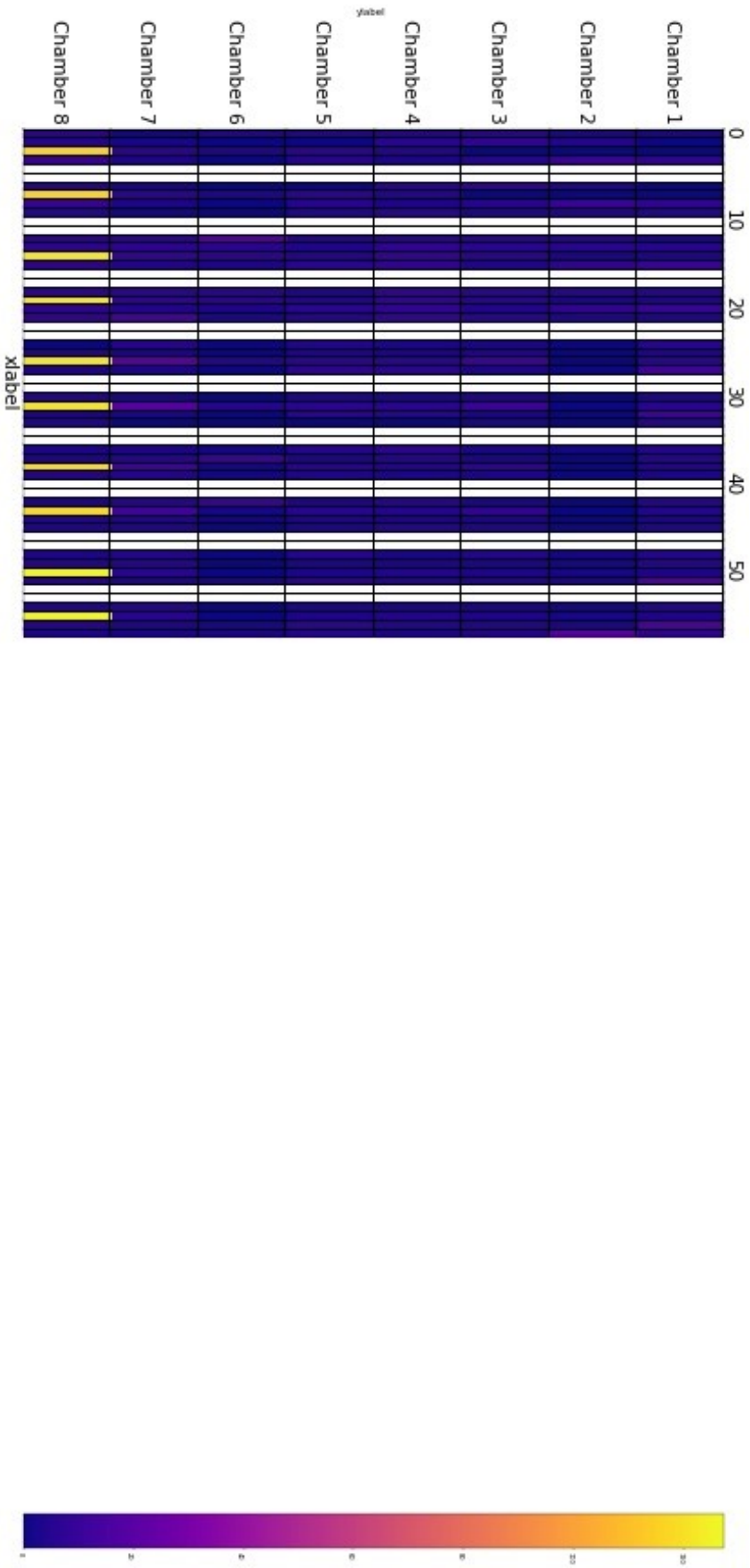


Figure E.4: Isolated Peak pattern

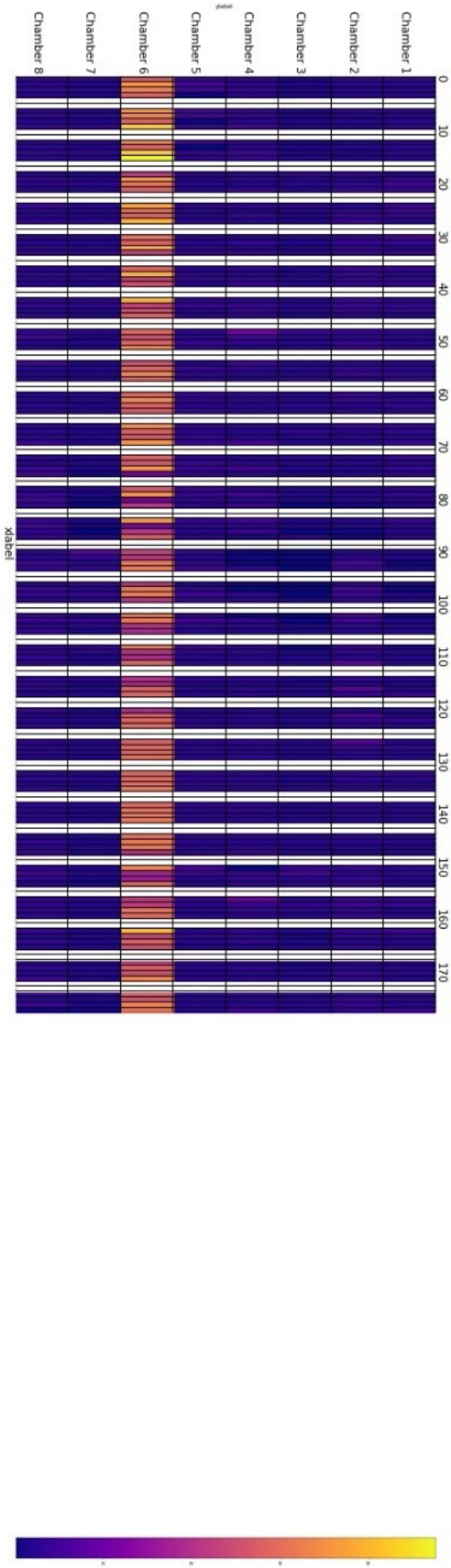


Figure E.5: Cyclic CO pattern

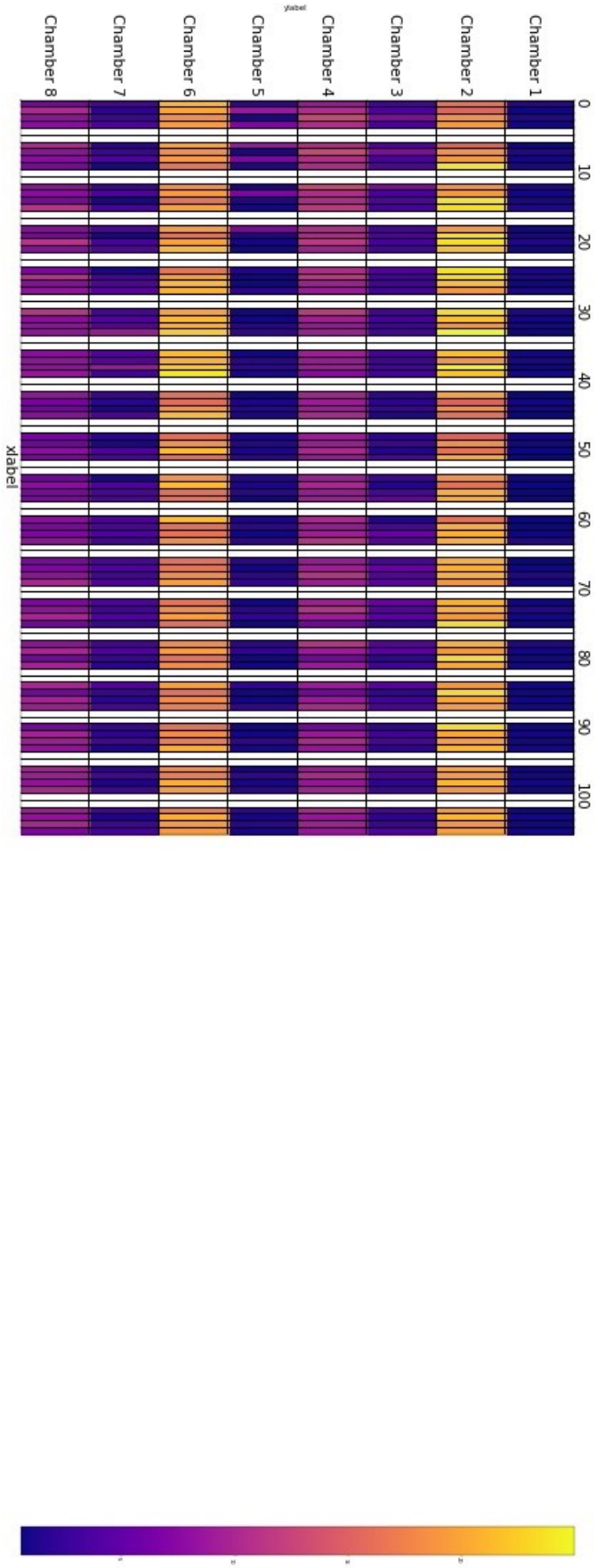


Figure E.6: Chamber CO associated to Cyclic CO pattern

Module	Framesize	eps	Nr. Clusters	% of Normal data	% of the biggest anomalous cluster
Unit 08	4	14,5	2512	89,61%	0,19%
	8	13,0	1700	81,58%	0,28%
	16	17,9	1002	80,27%	0,62%
Unit 29	4	12,2	1547	90,11%	0,59%
	8	22,8	658	93,93%	0,19%
	16	24,2	600	90,14%	0,32%
Unit 61	4	11,6	2696	83,75%	0,49%
	8	19,1	1122	89,78%	0,24%
	16	17,5	1217	80,74%	0,68%
Unit 62	4	11,6	2154	87,76%	0,35%
	8	19,1	801	93,62%	0,13%
	16	18,8	771	89,37%	0,31%
Unit 11	4	13,8	1760	88,10%	0,51%
	8	18,0	893	90,79%	0,19%
	16	20,0	779	86,69%	0,39%
Unit 17	4	11,9	1506	91,82%	0,61%
	8	9,0	2045	80,85%	3,53%
	16	11,5	1685	80,67%	0,57%
Unit 43	4	13,8	1599	90,95%	0,37%
	8	10,3	1916	82,20%	0,35%
	16	15,5	1090	84,25%	0,34%
Unit 14	4	12,1	1841	91,74%	0,39%
	8	9,1	2256	82,49%	2,21%
	16	18,2	934	90,40%	0,30%
Unit 27	4	14,2	1995	88,47%	0,60%
	8	19,7	911	92,26%	0,44%
	16	18,0	942	86,07%	0,82%
Unit 33	4	13,7	1472	91,48%	1,00%
	8	10,7	1792	83,68%	3,19%
	16	23,8	777	90,77%	0,54%

Table E.1: Results of application of the clustering technique in each module and each frame

Appendix F

Anomalous patterns found with secondary approach based on the 'PeakValue' variable's signal variation

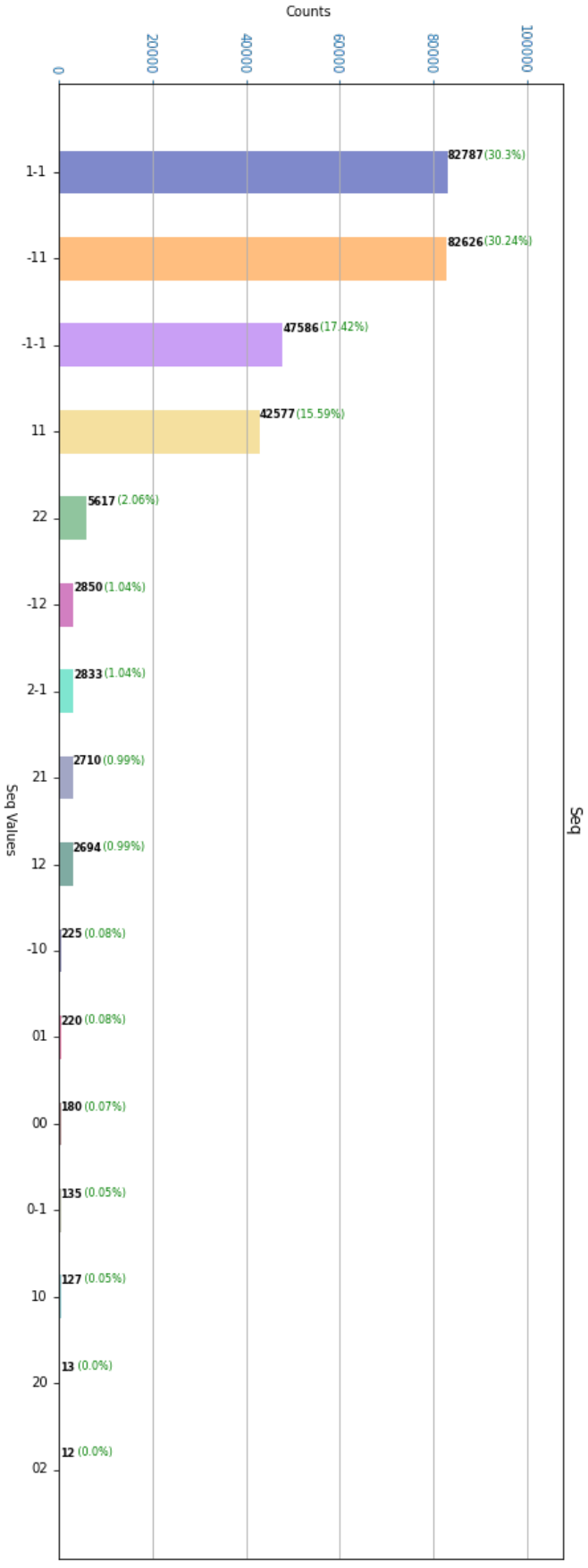


Figure F.1: Frequency of each type of variation in two consecutive cycles

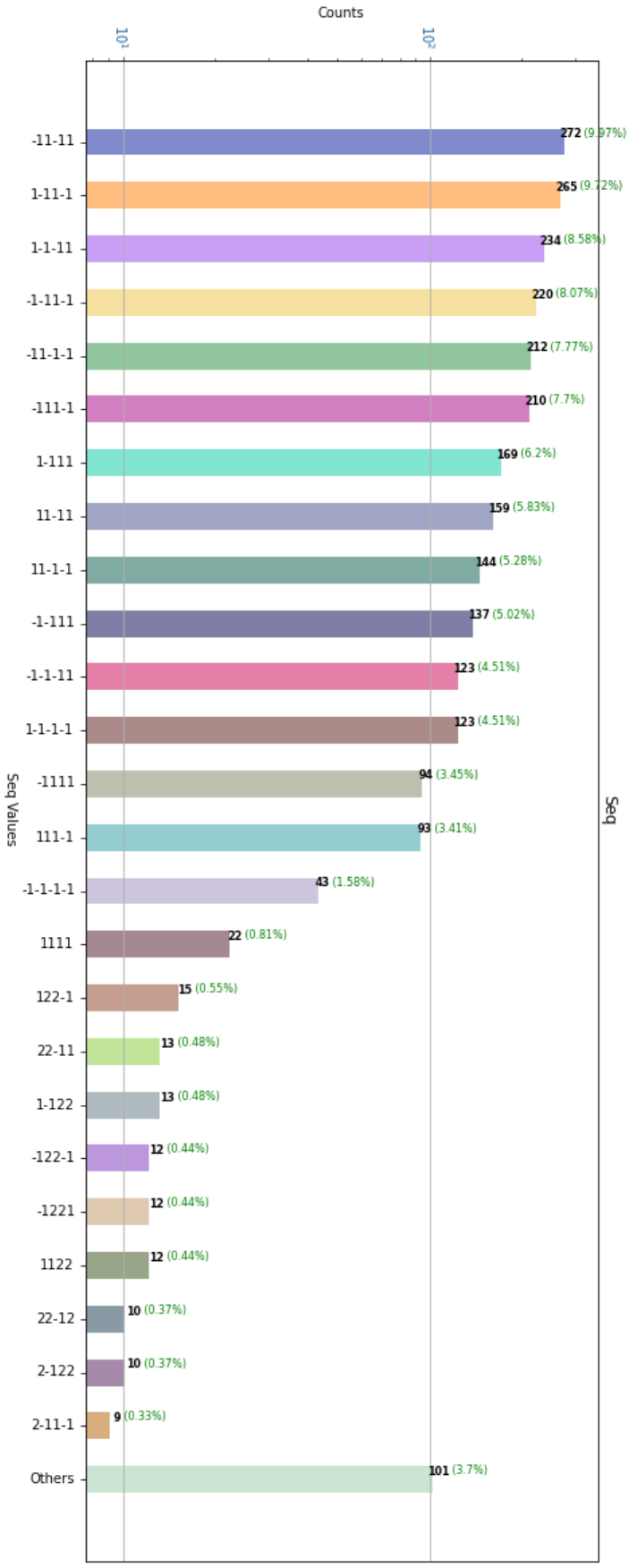


Figure F.2: Frequency of each type of variation in four consecutive cycles

Appendix G

Anomalous patterns found with secondary approach based on the 'Error' variable

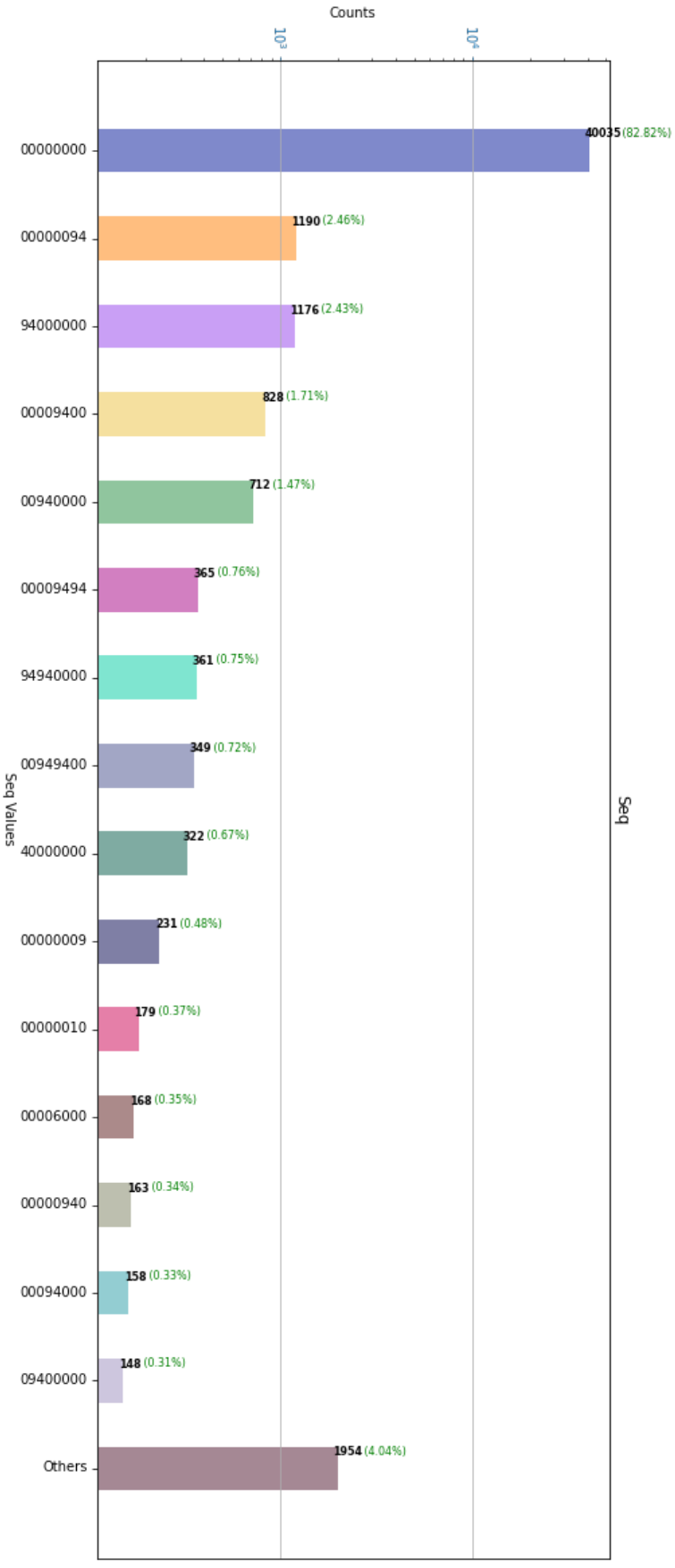


Figure G.1: Most common sequence of errors in one cycle

Appendix H

Selection of the main sample

Beginning of Table H.1		
Module	Score	Rank
08	0,2	1
61	0,3	2
29	0,3	3
62	0,3	4
54	0,4	5
12	0,4	6
36	0,4	7
38	0,4	8
48	0,4	9
51	0,4	10
04	0,4	11
49	0,4	12
37	0,4	13
55	0,5	14
22	0,5	15
25	0,5	16
53	0,5	17
39	0,5	18
31	0,5	19
23	0,5	20
18	0,5	21
40	0,5	22
02	0,5	23
34	0,6	24
15	0,6	25
26	0,6	26
24	0,6	27
57	0,6	28
43	0,6	29
11	0,6	30
17	0,6	31

Continuation of Table H.1		
Module	Score	Rank
47	0,6	32
10	0,6	33
16	0,6	34
05	0,6	35
21	0,6	36
50	0,6	37
03	0,6	38
09	0,6	39
45	0,6	40
13	0,7	41
30	0,7	42
41	0,7	43
60	0,7	44
35	0,7	45
58	0,8	46
59	0,8	47
06	0,8	48
19	0,8	49
01	0,8	50
46	0,8	51
42	0,8	52
44	0,9	53
27	0,9	54
33	0,9	55
14	0,9	56

Table H.1: Ranking of the Modules

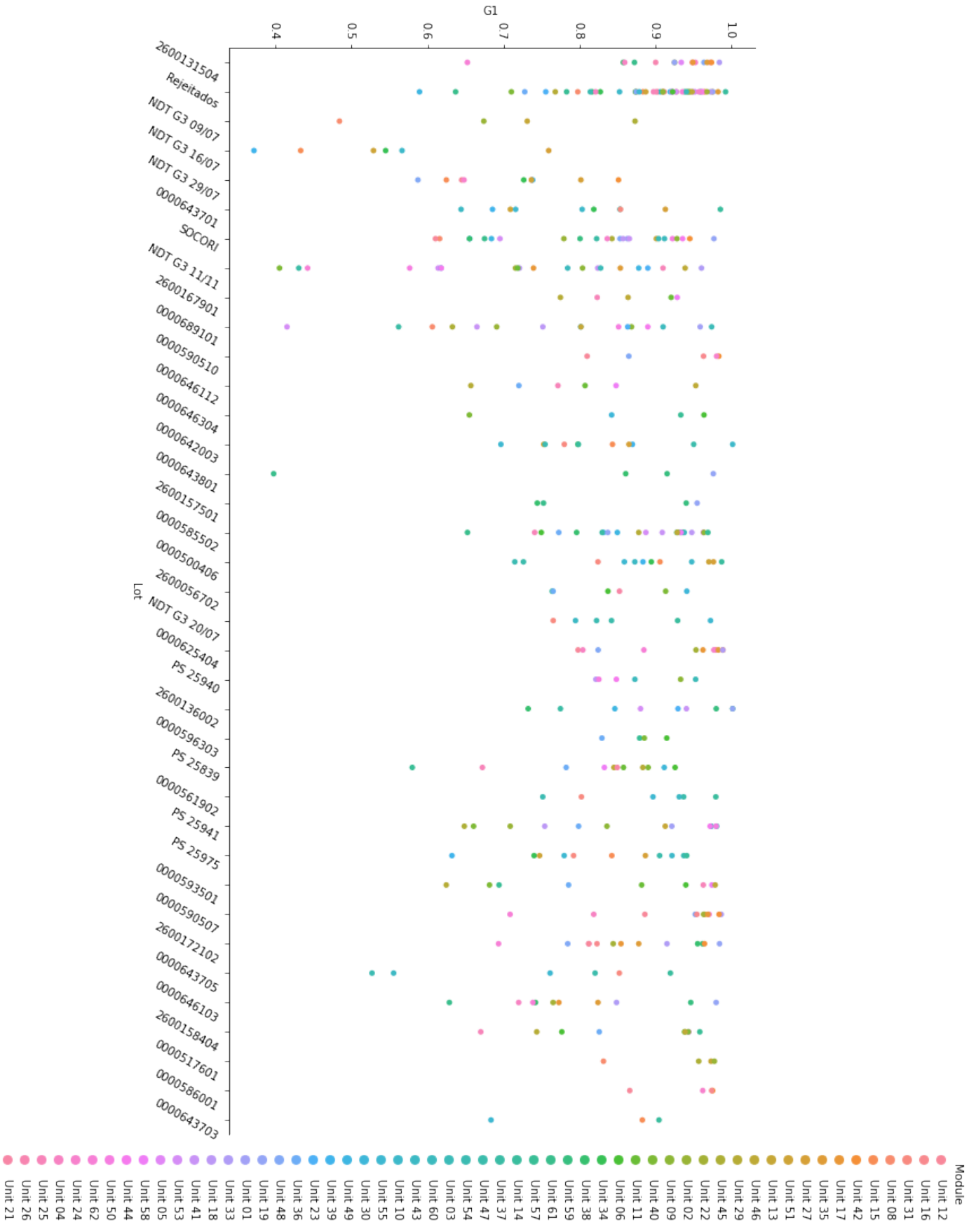


Figure H.1: Acceptance of a sample of lots by different modules

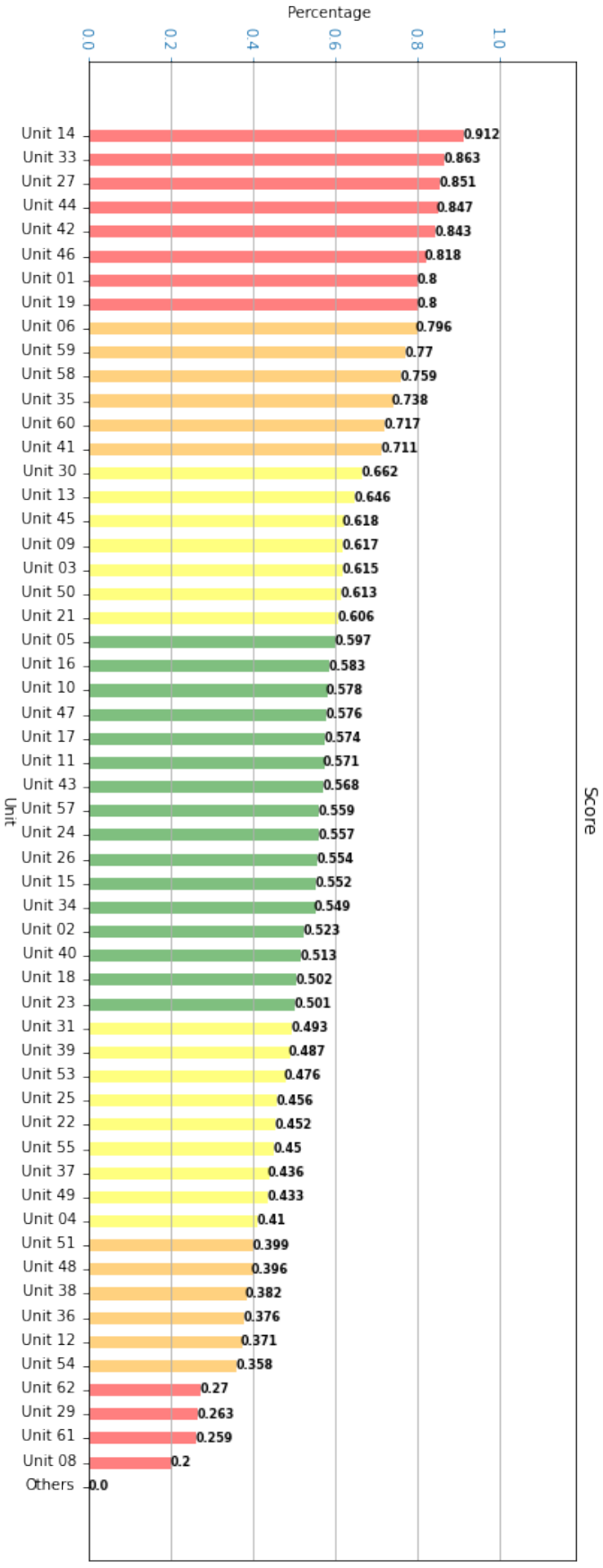


Figure H.2: Graphical representation of the module ranking