SCHOOL OF ECONOMICS AND MANAGEMENT
UNIVERSITY OF PORTO

AMORIM CORK

Anomaly detection algorithm in individually TCA analysis by NDTech 2.0

14 of September, 2023 | Gustavo Baldaia

U.PORTO
FEP FACULDADE DE ECONOMIA
UNIVERSIDADE DO PORTO

Implementation and validation of anomaly detection algorithm on NDTech, individually TCA detection technology on cork stopers

Gustavo Nuno Prata Baldaia

Project Report
Master in Data Analytics

Supervised by
Prof.º Drº Bruno Veloso, FEP

Co-supervised by
Prof.º Drº João Gama, FEP
Prof.º Dr.º Miguel Cabral, Amorim Cork S.A.

2023

Acknowledgments

The successful completion of this dissertation would not have been possible without the invaluable assistance of numerous individuals, both directly and indirectly involved in its development. I would like to extend my deepest appreciation to each and every one of them.

Firstly and foremost, my gratitude goes to Amorim Cork S.A., and in particular, my co-supervisor Miguel Cabral. His provision of essential resources and mentorship has been instrumental at every stage of this project, guiding it in the right direction. I am also thankful to the dedicated members of the NDTech team whose collaboration and contributions during their working hours greatly benefited this research and led to the conclusions presented in this report. Special thanks to the NDTech department managers, Carlos Costa and Joel Ferreira, for their warm welcome and invaluable insights throughout the project. Additionally, I want to express my gratitude to Rogério Nunes for his efforts to become an integral part of the project. His valuable perspectives and insights have enriched the project, leading to the development and improvement of the most appropriate choices.

Secondly, I want to express my appreciation to my FEP supervisor, Bruno Veloso, and co-supervisor, João Gama, for their guidance and invaluable insights, especially during the final stages of this thesis when working against tight deadlines.

My genuine thanks extend to my family and friends for their unwavering encouragement, understanding, and trust in my abilities. Their continuous support has been a constant source of motivation.

Abstract

Cork has become the preferred choice for sealing wine bottles, especially in high-quality wines produced from France, Italy, Spain, or Portugal. Due to cork stoppers' strong reputation in the wine industry, there is a substantial emphasis on ensuring efficient systems that guarantee both production capacity and product quality. This project, in collaboration with Amorim Cork S.A., proposes an online system to address the limitations related to human errors and effectiveness constraints faced by the existing human-dependent control system. The system should be able to identify anomalous behaviours from the state-of-the-art NDTech technology, a complex non-destructive gas chromatography screening system designed to detect undesired components, particularly TCA, in cork stoppers, from Amorim Cork S.A. The objective is to establish whether abnormal behaviour can be detected in real-time from NDTech operations, improving the existing control system. An LSTM autoencoder was chosen for its suitability with the sequential and unsupervised data provided by NDTech 2.0 operation. In this report, was conducted a thorough analysis of the data, including a comprehensive study of intentionally induced malfunctions simulations under a controlled environment. Through the utilization of the reconstruction error from the optimized trained models and the implemented adaptive threshold, it's possible to determine whether each instance falls within the normal operation. The model's performance under the controlled simulations were overall positive, where they could identify anomalies within most of the malfunctions. However, due to the absence of labels it was not possible to conclusive extrapolate this performance to the online tests. To support the team on-site with real-time information regarding the anomaly detection results, a conceptual visualization tool was introduced. This tool has potential to be instrumental in verifying the accuracy of detected anomalies and enabling a more proactive response from the team to promptly address potential issues.

Key words: NDTech 2.0**,** Anomaly detection, LSTM autoencoder, Unsupervised system, Streaming data

Resumo

Atualmente a cortiça é a escolha preferida para selar garrafas de vinho, especialmente quando abordados os produtos de alta qualidade produzidos por reconhecidos produtores de França, Itália, Espanha ou Portugal. Esta valorização atribuída pelo mercado, obrigou a um forte investimento em sistemas eficientes que consigam garantir tanto a capacidade de produção como a qualidade do produto. Com a colaboração com a Amorim Cork S.A., é proposto neste projeto um sistema online capaz de combater as atuais limitações do sistema de controlo existente dependente da intervenção humana. O objetivo deste trabalho é perceber se este sistema é capaz de idenficar comportamentos anómalos do NDTech, uma tecnologia de ponta de análise individualizada da presença de TCA nas rolhas de cortiça através da cromatografia, desenvolvida pela Amorim Cork S.A., apartir dos dados registados da operação da mesma. O algoritmo LSTM autoencoder foi escolhido devido à sua adequação aos dados sequenciais e não supervisionados fornecidos pelas operações do NDTech 2.0. Foram realizas análises minuciosas aos dados, incluindo o estudo de avarias intencionalmente provocadas num ambiente controlado. Através da combinação dos erros de recontrução obtidos pelos modelos treinados e optimizados com os limites adaptatvivos implementados, foi possível identificar se cada uma das observações encontra-se dentro do funcionamneto normal esperado. Embora o desempenho do sistema tenha sido globalmente positivo analisando os períodos das simualações, devido á ausência de qualquer tipo de classificação dos dados, não foi possível concluír o mesmo durante os testes online. Com o objetivo de apoiar a equipa no terreno, foi introduzida uma ferramenta de visualização conceptual com acesso em tempo real à informação dos resultados obtidos pelo Sistema, que demonstrou potencial tanto no papel de verificação da precisão das anomalias detetadas como na facilitação de uma resposta mais proativa da equipa para resolver prontamente potenciais problemas.

Palavras Chave: NDTech 2.0**,** Deteção de anomalias, LSTM autoencoder, Sistema não supervisionado, Sistema online

INDEX

Abstract ii
Resumo iii
Figure List vii
Table List ix
Chapter 1 Introduction 1
1.1 Introduction 1
1.1.1 The Cork Industry 2
1.1.2 TCA presence on cork 2
1.1.3 NDTech 3
1.1.4 Machine Learning and anomaly detection 4
1.1.5 Structure of the project 4
Chapter 2 Literature Review 7
2.1 Anomaly detection 7
2.1.1 Anomaly detection on unsupervised systems 8
2.1.2 Autoencoders 10
2.1.2.1 Long Short-Term Memory Autoencoder 11
2.1.3 Streaming Environment 13
2.1.4 Summary 14
Chapter 3 Methodology 17
3.1 Problem Description 17
3.1.1 NDTech 2.0 18
3.2 The Data 20
3.2.1 Exploratory Data Analysis 22
3.2.1.1 Distribution 22
3.2.1.2 Missing Values 25
3.2.1.3 Outliers 26
3.2.1.4 Data Scalling 29
3.2.1.4.1 Online Data Scalling 29
3.2.1.5 Correlation 30

3.2.1.6 Kernel PCA 31
3.3 The Model 34
3.3.1 LSTM Autoencoder 34
3.3.1.1 Algortihm Training 35
3.3.1.2 Algortihm Optimization 36
3.3.1.3 Threshold 38
3.3.1.3.1 Online Threshold 38
Chapter 4 Experimental Results 41
4.1 System Architecture 41
4.2 NDTech 2.0 Malfunctions 43
4.2.1 Simulations 43
4.2.2 Anomalies Detected 49
4.3 Online Environment 53
4.3.1 System Architecture 54
4.3.2 Results 56
4.3.2.1 Results visualization 61
Chapter 5 Final Considerations 65
5.1 Conclusions 65
5.2 Future Considerations 68
Chapter 6 References 69

FIGURE LIST

Figure 2.1 - Structure of the different anomaly's detection frameworks (Source: Goldstein et al., 2016) 9
Figure 2.2 – Architecture of LSTM network module (Source: Nguyen et al., 2021) 11
Figure 2.3 – Architecture of an LSTM autoencoder model, with the structure of the network for encoding and decoding the input sequence (Source: Mobtahej et al., 2022) 12

Figure 3.1 – General layout of a processing unit within NDTech 2.0 system 19
Figure 3.2 – List of NDTech 2.0 variables 21
Figure 3.3 – Violinplots of position variables for the distinct valves (LI, SS, DS, CS) by processing unit 23
Figure 3.4 – Histograms of the average range of values for the Box feature *(Box104, Box105, Box106, Box 107, Box108)* by chamber for Punit 3 24
Figure 3.5 – Boxplots of the variable DS_Torque by chamber group 25
Figure 3.6 – Bar plot with the total number of identified outliers with IQR method for each chamber group within each PUnit 27
Figure 3.7 – a) Bar plot with the total number of identified outliers with Z-Score method for each chamber within each Punit; b) Bar plot with the distribution of the outliers from PUnit 3 for each chamber group across the variables 28
Figure 3.8 – Correlogram for PUnit 15 31
Figure 3.9 – Kernel PCA 2D representation by PUnit 32
Figure 3.10 - Kernel PCA 2D representation by chamber group 32
Figure 3.11 - Kernel PCA 2D representation by chamber from all the PUnits 33
Figure 3.12 – a) Plot with models training loss with the different length datasets for the odd chambers of PUnit 10; b) Plot with models training and validation loss for the 60 and 90-days datasets for the odd chambers of PUnit 10 35

Figure 4.1 – Chart flow of the anomaly detection system 42

Figure 4.2 – Plots with the average's values of the *Box* (*B105, B106, B106, B108*) variables from both chamber groups of PUnit 15 during the surrounded period of an induced malfunction on the nitrogen injection to the system.....45
Figure 4.3 - Plots with the average and maximum values of the *Chamber_Pressure* variable from the odd chambers from PUnit 10 during the surrounded period of an induced malfunction in the chamber 1 of the PUnit 10 46
Figure 4.4 - Plots with the average and maximum values of the *Chamber_Pressure* variable from the odd chambers from PUnit 15 during the surrounded period of an induced malfunction in the chamber 3 and 5 of the PUnit 15 46
Figure 4.5 - Plots with temperature registered from the incubator chambers, columns and detector for both chamber groups of PUnit 12 during the surrounded period of an induced malfunction in the PUnit 12 47
Figure 4.6 – Pie chart with the count of malfunction with and without reflections on the data. 48
Figure 4.7 – Plot with the model's reconstruction error for the odd chambers from PUnit 10 during the surrounded period of an induced malfunction in the chamber 1 of the PUnit 10 50
Figure 4.8 - Plot with the model's reconstruction error for the odd chambers from PUnit 10 during the surrounded period of an induced malfunction in the chamber 3 and 5 of the PUnit 15 50
Figure 4.9 - Plot with the model's reconstruction error for chambers groups from PUnit 15 during the surrounded period of an induced malfunction on the nitrogen injection to the system 51
Figure 4.10 - Plot with the model's reconstruction error for chambers groups from PUnit 15 during the surrounded period of an induced malfunction on the initial position from the Stream Selector valve. 52
Figure 4.11 - Chart flow of the online anomaly detection system 55
Figure 4.12 – Plots with the model's reconstruction error for some of the subsystems from the online tests.....58
Figure 4.13 - Plots with the model's reconstruction error for the odd chambers from PUnit 15 for a selected interval from the online tests 59
Figure 4.14 - Plots with the minimum, average and maximum values of the *Chamber_Pressure* variable from the odd chambers from PUnit 15 during the previously selected interval from the online tests.....60
Figure 4.15 – Conceptual visualization window for the results from the online AD 62
Figure 4.16 - Conceptual dashboard with a list of the anomaly detected from the online AD.....63

TABLE LIST

Table 3-1 – Table with examples of observations with missing values in the *SS_Torque_Min* variable 26
Table 3-2 – Table with the range for all parameters considered during the optimization 37

Table 4-1 – Number of observations under conditioned operations 53
Table 4-2 – Table with the summary of the results from the online test 57

CHAPTER 1 INTRODUCTION

An introduction of the context of the proposed project, the challenge faced, the goals and its planification.

1.1 INTRODUCTION

Today with the fast and huge development of technology, the access to data allowed new perspectives on how problems and processes can be solved or improved. Data analysis and machine learning are being implemented in the majority of businesses and systems in order to achieve better and most reliable outcomes, and quality control proceedings are not an exception to this phenomenon. Nevertheless, there is still space to improvement, and remain a challenge to transform all the data collected into value information to help make decisions. There is still a heavy reliance on human interpretation and knowledge, as is still hard to generate useful and reliable decision support information from the retrieved data (Ji, & Abourizk, 2018).

In this report will be study the possible benefits of the implementation of an automated process to the NDTech technology that was developed by Amorim Cork, S. A., with the ambition to develop a system capable of detecting operating malfunctions on the production line of NDTech. This technology focuses on the individual analysis of corks to detect the presence of a significant contaminant that can impact their quality, namely 2, 4, 6 – Trichloroanisol (TCA).

This project builds upon the previous work performed by Rogério Nunes, regarding the conversion of the operational data from NDTech 2.0 records into valuable information to detect any malfunctions in the technology (Nunes, R., 2022). The main goal is to investigate whether abnormal operations within NDTech can be effectively identified by a system relying only on data generated by the technology itself. Additionally, the objective is to understand if this system can help improving the current control system. To this matter, a broad analysis of the data is central. This analysis can provide valuable insights that can play an important role for the development of the online anomaly detection system capable of analysing registered operational instances in real-time. In addition, to further extent the current knowledge, controlled malfunctions will be deliberately induced into the NDTech operation to test how these anomalies evidence in the data. These simulations will help fine-tuning and validating the developed system's capabilities in identifying some of the various types of anomalies within the NDTech technology.

1.1.1 THE CORK INDUSTRY

The cork industry had always stood as one of the oldest and most sustainable industries in the market, due to the unique and remarkable qualities of cork. Before the beginning of a successful and close collaboration with the wine industry, the Egyptians and Persians first used cork to produce fishing floats, followed by the Ancient Greeks and the Romans that used it to make sandals and to seal amphora jars. Since the late 1700s cork stoppers became the closure of choice for wine bottles and represent the major portion of the focus of this industry with almost 13 billion cork stoppers produce every year. Nowadays with the increased environmental awareness and the exploitation of cork qualities, new markets and applications are emerging (Leithead, 2022).

Cork remains a desirable product, as the industry experience a continuous steady growth already playing an important economic impact both locally and globally. Portugal as the largest producer of cork, covers nearly half of the global production followed by cork-producing countries including Spain or Italy.

The role played by Corticeira Amorim as the emerged world's leading producer and cork products supplier can't be ignored. The group founded in 1870, lead ever since the course to promote and elevate the cork potential, supported by a very strong commitment and investment in innovation, sustainability, and quality. The pioneered introduced cutting-edge technologies and sustainable processes that have been revolutionized the industry and played a critical role in increasing the demand for cork products worldwide as well as attracting new sectors.

1.1.2 TCA PRESENCE ON CORK

Wine contamination or taint is an important problem that always concerned winemakers, and although the presence of undesirable compounds on wine can have different sources, since the eighties and nineties, cork stoppers became the most commonly blamed for this event, which is known as "cork taint" (Rodríguez et al, 2009). These beliefs were leveraged over the past decades by the stablish contribution of several compounds that can contribute to the cork taint, where among them, TCA plays a major role (Tarasov et al, 2022). This organic compound is known by its capability to

taint the aroma and taste of wine, producing what is commonly describe as a musty or mouldy odour, that can negatively affect the sensory experience of wine or even make it undrinkable.

The cork industry, consequently, face nowadays, strict quality standards from the market, where the final aim is to produce a product with a total absence of any kind of defects that don't have a negative impact on the quality of the wine and that can result in in substantial financial losses. The Cork industry have shifted its focus into this challenge in the early days of this century, investing considerable amounts of time and money developing procedures that can successfully prevent, reduce, and detect such compounds from the natural cork. Among these vast list of innovations systems, is NDTech's TCA detection technology, from Amorim Cork S.A. that will be the pillar of this project.

1.1.3 NDTECH

The ground-breaking system developed by Amorim Cork S.A. in collaboration with a British company from Cambridge, an expertise in chromatography, addresses the important TCA problem faced by the industry by taking advantage of cutting-edge scientific and technologic developments in combination with the current knowledge and techniques being used. Amorim's NDTech technology can and have change how the guaranteed quality of cork was assured.

The TCA detection system combines a non-destructive gas chromatography screening system that is capable of precisely analyse each cork stopper without causing any structural damage to it. The complex system developed, reduced the current extensive analysis time of 15 minutes from traditional methods to around 20 seconds per cork stopper, which is a remarkable achievement with the preservation of precision and quality of the process.

This game-changing technology sets a new standard for the market, addressing the limitations of traditional quality control methods, which rely on subjective human sensory analysis. It also offers a better alternative to the typically time-consuming and often destructive approaches, while maintaining high reliability and accuracy. Amorim through NDTech open the door to provide the market with TCA-free corks, safeguarding the integrity and flavour profiles of the wines they seal, with an individual, non-destructive analysis that allow a fast and precise identification of contaminated corks.

1.1.4 MACHINE LEARNING AND ANOMALY DETECTION

With the current market's demand, the efficiency of processes is crucial to any business, the errors are becoming less acceptable, and the identification of unusual events or deviations from the expected behaviour is crucial to recognize potential problems in any type of system. Anomaly detection problems are applicable across a broad spectrum of domains and scenarios. They find utility in manufacturing facilities, helping to identify, and segregating faulty products, as well as in predicting equipment failures, maintaining quality control, and enabling proactive maintenance. Additionally, in finance where they help in the early detection of anomalies related to financial transactions or market behaviours. Moreover, in the field of cybersecurity, anomaly detection is vital for spotting network intrusions and malicious activities. These are just a few examples of the numerous environments where anomaly detection proves invaluable.

The detection of abnormal observations is a challenge that have been studied for a long time, where a variety of approaches have been developed and used to solve this problem. However, with the increasing complexity and the evolving nature of data available, new techniques are emerging and growing in significance, such as Machine Learning (ML) algorithms, that already play an important role in this area (Nassif et al, 2021). In both supervised environments where there are available labels of the data, as well as in the cases where this information is not accessible ML methods excel in identifying deviant data, by capturing the most relevant information from the learned representations and boundaries.

1.1.5 STRUCTURE OF THE PROJECT

This report comprises 4 additional chapters, each representing distinct developing stages of the project introduced above. These phases can be summarized as follows:

- ***Literature Review*** – where will be conducted a study and acknowledgement of the presented theories and analyses that can support and sustain the diverse approaches followed during the development of the report.

- ***Methodology*** – encompassing the introduction of the problem description, the data and the model selected for the problem. The former will include the detailed presentation of the proposed project including a more detailed description of the NDTech 2.0 technology and

process. The second will cover a brief description of the data available as well as an exploratory data analysis of the underlying data. Finally, the third will introduce the model selected for the project and its development, training, and optimization.

- ***Experimental Results*** – covering the system architecture designed to the problem, the study of the induced malfunctions including the models reaction to them and the anomalies detected under those periods. Finally, the implementation of the online system will be introduced with the required adaptations for the system's architecture and the analysis of the results obtained from a 15-day period tests. This chapters culminated with the introduction of a conceptual visualization tool with real-time information from the system's results.

- ***Conclusions*** – containing a summary of the achievements obtained from the development of the project, the main barriers faced and possible future improvements or ideas to be explored.

CHAPTER 2 LITERATURE REVIEW

An analysis of existing research and work related to the studied topic, in order to situate this project to the existing knowledge.

In this chapter will be reviewed different existing approaches and techniques applied to this project's topic focusing on the anomaly detection area and some underlying requirements that may be crucial for its success.

2.1 ANOMALY DETECTION

The rapid advance of network technologies helped to exponentially connect all the corners of the globe together, which unfolded an extensive new world of ways to interact with each other, including contact with enterprises. Although the opening of excited and massive advantages that came along with the progress, it also revealed accordingly grown of concerning security threats that propelled the development of intrusion detection systems. This fact produces a common association of anomaly detection with crime and security related problems, although in fact there are numerous in such cases, as fraud detection on credit cards, insurance or cyber security intrusion detection and military surveillance for enemy activities. Nowadays the range of areas and problems where this type of algorithm can positively impact systems operability is extensive, from manufacturing to health care. (Pang et al, 2021 and Provotar et al., 2019).

Anomaly detection can be described as the search and recognition of behaviours that do not follow the defined 'normal' or expected pattern notion of a particular system. These dubious observations or behaviours are commonly referred as anomalies or outliers and can arise from a variety of sources, from external sources, as malicious activities or malfunctions of the system depending on the domain of the topic studied (Varun et al., 2009).

Some topics usually related to anomaly detection are noise removal, noise accommodation and novel detection. The former two procedures are meant to deal, in different manners, with noise presence on data, which are the inclusion of unwanted objects, that can be an obstacle, leading to inadequate approaches and misleading conclusions. Novel detection differs from anomaly detection, as the

emergent unknown patterns have an underlying reason to occur, and later are typically acknowledged as part of the normal behaviour of the system. What often is observed is that these related problems, often shared solutions among them (Varun et al., 2009).

The study of these abnormal presence on data, go back from as early as the 19th, by the statistics community (Edgeworth, 1887), where different anomaly detection approaches have been developed and studied for specific domains as well as for more generic applications. The recent uprising of machine learning developments managed to make it play a major role in anomaly detection solutions (Bhuyan et al. 2014). Although this is a well-studied problem, due to its significance, it remains a challenging task to build efficient anomaly detection algorithms for complex and muti-dimensional data, due to the many factors that hamper the process. The main challenges faced in this type of problems can be summarized as follows (Ahmed M et al. 2016, Varun, 2009):

- Defining an accurate normal region that incorporates every single normal object of the data is a very difficult problem, as usually the boundaries between normal and abnormal behaviours are often minimal. In addition, in many systems, the normal behaviours are constantly evolving and changing over time, so the notion of normal objects must evolve as well.
- The particularities of complex systems restrict the possibility of applying universal techniques, which demands domain focused approaches.
- The presence of noise in the data tends to have a similar behaviour as an anomaly, enhancing the difficulty of distinguishing them.

2.1.1 ANOMALY DETECTION ON UNSUPERVISED SYSTEMS

Depending on the information available regarding the data in use, the anomaly detection setup to be applied can differ. The types of setups can be distinguished between three main frameworks, supervised, semi-supervised and unsupervised algorithms (*Figure 2.1*). The former two setups demand more knowledge and insights of the data, where the anomalies should be identified, in different manners, prior to the implementation of the system. Both use a train and test data set approach, differing on the type of data that each of them contains. On supervised models, both data sets are fully labelled, similarly to a traditional classification/pattern recognition problem, where in this case, the data is typically highly unbalanced. Nowadays, with the amount of information and data retrieved from any technological systems, it is most likely to be an unfeasible task to fully label

observations, narrowing the applicability of these models (Goldstein et al., 2016). In semi-supervised systems, the training data set must not contain any anomaly observations, to be able to develop the ability of recognizing the objects that deflect from the normal pattern defined on the test data set. Unsupervised techniques, on the other hand, require no labelling of the data, presenting more flexible models as there is no distinction between training and testing data sets and the algorithm can be applied on the raw data.

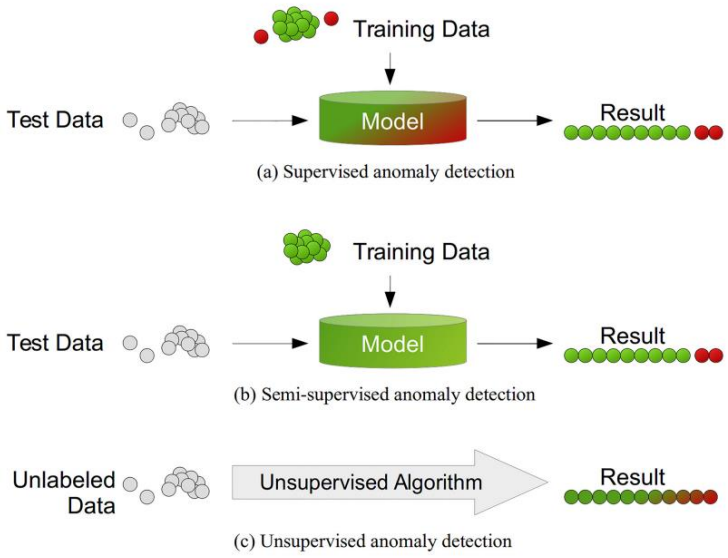


Figure 2.1 - Structure of the different anomaly's detection frameworks (Source: Goldstein et al., 2016)

The other main distinction from the unsupervised anomaly detection setup from the former two, is the type of output generated. Both supervised and semi-supervised algorithms typically return a labelled output, where the objects are either considered normal or abnormal. Alternatively, a score or confidence levels that ranks the observations into a likelihood of being considered as outliers, is the most used output for unsupervised environments. For this reason, the evaluation of the performance quality of these approaches is also distinct. When facing a classification problem, with labelled outputs, Receiver Operating Characteristic (ROC) or Precision-Recall (PR) curves are the preferred choices to evaluate and compare algorithm's performance. For unlabelled data, these metrics cannot be applied, and there isn't a direct method to estimate and evaluate the algorithm performance. However, it is possible to compare the results from different methods, and the two common applicable criteria, reviewed and presented in the literature are the Mass Volume (MV) and the Excess-Mass (EM) (Goix et al., 2016).

Different metrics can also be used as useful performance measurements when comparing different algorithms for the same task or challenges, as the comparison of execution time, training time when required and also some benchmarks datasets that attempt to recreate the real world as reliably as possible (Torralba and Efros, 2011), permitting to compare the performance of different algorithms that solve wide-ranging problems, as the Yahoo Webscope benchmark or Numenta Anomaly Benchmark among others.

The numerous techniques available for anomaly detection problems cover very distinct approaches and perspectives of how to identify the desirable result, although there is a clear distinction between parametric and nonparametric algorithms. The former relies on a predefined number of fixed parameters to model the data, while nonparametric algorithms can in theory, have an infinite number of parameters that is dependent on the complexity of the data. Parametric algorithms are for that reason, typically faster, but demand some insights of the data to obtain a proper performance, on the other hand, the latter type of algorithm requires less knowledge from the data studied (Russell et al, 1995).

2.1.2 AUTOENCODERS

Autoencoders are a type of artificial neural network that is trained to reconstruct the input provided. They were first introduced back in 1986 by Hinton and its group (Rumelhart et al., 1986). They are primarily used to learn in an unsupervised manner to transform the input into an informative and compressed representation, that later can be reconstructed back as well as possible into its original shape.

A general formulation for an n/p/n autoencoder model as detailed by Baldi, 2012, is compose by a class of functions A: $\mathbb{R}^n \rightarrow \mathbb{R}^p$, or encoder, and B: $\mathbb{R}^p \rightarrow \mathbb{R}^n$ or decoder, that transforms an input vector $x \in \mathbb{R}^n$ into an output vector $A \circ B(x) \in \mathbb{R}^n$. The model problem is to learn the functions that can minimize the deformation function given by,

$$\arg \min_{A,B} \mathrm{E}(\Delta(A \circ B(x), x)$$

where E, is the expected distribution of x by the trained model, and Δ is a dissimilarity or distortion function, that measure the reconstruction error from the output of the decoder and the input. This

reconstruction error is the distance between the model representation and the original object (Baldi, 2012).

Numerous potential derivations from this general framework can be developed based on the goals of the problem, changing the different parameters of the structure. Autoencoders have been applied with success in several domains and applications including classification problems, data generation, clustering, anomaly detection or dimensionality reduction (Bank et al., 2020). The unsupervised learning capability of this model makes them exceptionally useful on unlabelled environments, a proven powerful and versatile tool in the field of deep learning. With the increase investment, research, and development in this field, autoencoders alongside with other machine learning techniques place themselves in the front row of the expected models to prevail and increase importance as the solution to complex real-life problems (Li et al., 2023).

2.1.2.1 LONG SHORT-TERM MEMORY AUTOENCODER

Long short-term memory Networks (LSTM) are a form of a recurrent neural network (RNN) that is most suitable for sequential regression problems. The capacity to address the vanishing or exploding explosive gradient problems common in other deep networks due to its unique memory and forgetting mode allows its flexible adaptability to timing characteristics of network learning tasks. (Wang et al., 2021).

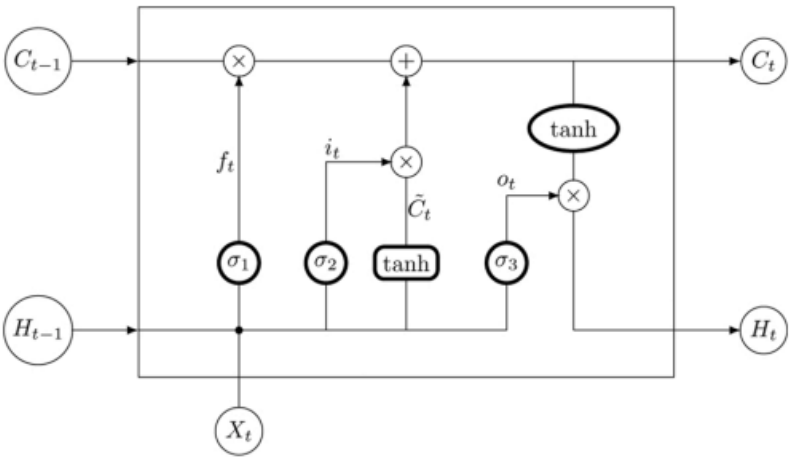


Figure 2.2 – Architecture of LSTM network module (Source: Nguyen et al., 2021)

The LSTM is an improved version of a basic RNN, where are added a memory cell throughout the time series, that its purpose is to store and estimate the important information that should be retained and discarded. For this task, each LSTM module (*Figure 2.2*) is composed of an input gate (i_t), a forgetting gate (f_t), an output gate (o_t), and a memory gate (C). Each of these gates, with the exception of the memory gate, is composed of a sigmoid function that defines the portion of the information that should be passed on. Firstly, the forgetting gate estimate the content of previous memory cell state C_{t-1}) that should be discarded, through the sigmoid function (σ_1). The input gate determines what information of the input vector (X_t) should be kept in the memory gate, in this case through the combination of the sigmoid (σ_2) and the tanh function. Lastly the output gate decides the slice of the memory cell that should be outputted into the next hidden state (H_t), having as an input, the input vector, the previous hidden state (H_{t-1}) that are passed through the sigmoid activation function (σ_3) and the current memory cell state (C_t) that is passed through a tanh function (Wang et al., 2021 and Nguyen et al., 2021).

The combination of autoencoders with LSTM networks, can be defined as a hybrid model, where the encoder and decoder layers of the autoencoder are the LSTM networks, as pictured in *Figure 2.3*. This approach combines the abilities of both methods, making it capable of minimizing the reconstruction errors of the reproduced input, while capturing patterns and dependencies in sequential data over time, due to the LSTM layer's capacity to retain and propagate information over sequences. Therefore, efficient applications of such models have been proposed for time-series prediction and anomaly detection (Nguyen et al., 2021, Pereira et al., 2018 and Principi et al., 2019).

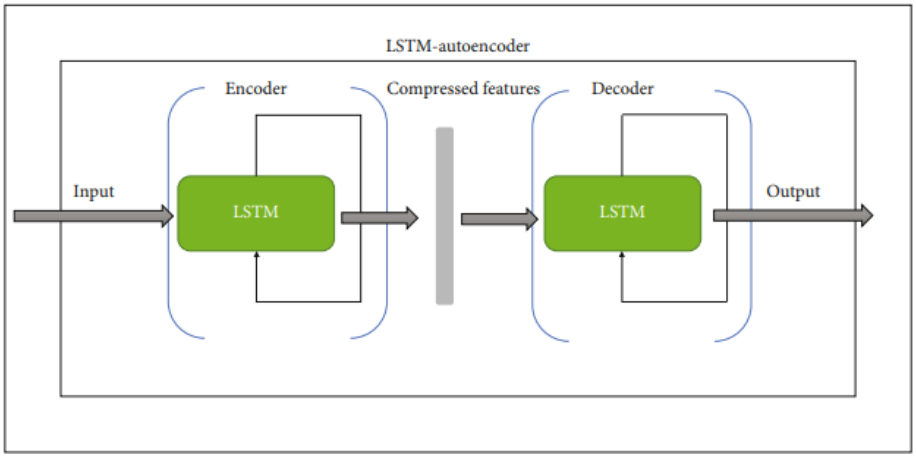


Figure 2.3 – Architecture of an LSTM autoencoder model, with the structure of the network for encoding and decoding the input sequence (Source: Mobtahej et al., 2022)

2.1.3 STREAMING ENVIRONMENT

Nowadays many are the technologies and applications capable of generating continuous in real-time data, which create new challenges and constraints that need to be dealt with. Data streams or streaming data are potentially infinite sequences of information and new instances that are in constant change and evolution. Streaming data is therefore impossible to store, and to be analysed as a whole, in the matter that common traditional limited data set problems are performed, due to the volume of data, and the continuous increase of information available (Thakran, T. & Toshniwal D., 2012).

Real-time streaming data analysis asks for non-traditional approaches to deal with the unique challenges faced, that are not present in batch processing where the whole dataset is available during the entire process. From handling each data record as they arrive, preprocessing it and analyse it, new constrains are introduced that require different solutions from batch applications. Streaming applications demands an algorithm capable of adapting to the changes that may occur on the data, and capable of identifying the underlying new patterns that may be developed from the often-non-stationary statistics of the data (Ahmad et al., 2017). Another important aspect that covers many data streams-based problems is the difficulty labelling all the existing data, which leads in most cases to unsupervised environments requiring algorithms that can be used in such systems.

In contrast to batch applications, there are different approaches to tackle and analyse the new data generated in online problems. Depending on the volume, variety, and velocity of the data (Gomes et al., 2019) may be unfeasible to individually process each new instance as they arrive, as other techniques such a sliding window or micro-batches, where the instances are collected into limited batches, that can either be set to a fixed length or cover all the data during a specified time period, that can be analysed jointly (Rettig et al., 2019). Feature scaling is another trivial procedure when the data statistics are known for the entire dataset, but scaling data streams comes with its own set of challenges. The nature of the problem requires adaptative and online techniques that can incorporate new observations and update in real time the scaling parameters. Scaling the data can be a critical preprocessing step for machine learning applications where the learning process rely on gradient descent to a faster convergence of the learning phase. The most commonly used approaches are the *Z-score* normalization, where the data is subtracted by the mean and divided by the standard deviation, $\frac{X-\mu}{\sigma}$, or *Min-Max Scaling* where the data is subtracted by the *min* value and divided by the range, $\frac{X-X_{min}}{X_{max}-X_{min}}$. While for the *Z-score* normalization for landmark approaches, is possible to achieve

by storing few information regarding the data, the adaptation for changing data patterns is slow which can lead to skewed results and suboptimal model performance. In sliding or moving windows approaches, which can be more appropriate to deal with a faster adaptation to changes, to calculate the exact values of the mean and standard deviation can become very computational demanding as it requires to store all the instance inside the current window (Gomes et al., 2019). For *min-max* scaling the difference between the data approaches is not as computationally impactful as the former technique, as both adaptative approach, where the minimum and maximum value is updated with the new coming data, or with fixed values, require little information to be stored.

2.1.4 SUMMARY

Anomaly detection is not a particularly new topic as it being explored for a long time, nevertheless it has expanded its importance within the current expanding technological world and data availability. It has proved to be helpful in leveraging the data usability to enhance productivity and effectiveness. The literature review regarding it is following the trend and more papers are being introduced with different approaches and perspective on the problem, namely machine learning techniques that have been growing in popularity lately.

From the reviewed literature while there were found various research on supervised, semi-supervised and even hybrid approaches combining supervised and unsupervised techniques, was found a notable gap on papers focus on pure unsupervised anomaly detection systems. The absence of labels on the data is a common challenge for complex and sophisticated systems that generate high volumes of data where is impossible to obtain a classification label for the registered records. Another limited topic where the literature review highlights the need for more research are online scaling techniques. Efficiently handling the increasing volume of data presents a substantial challenge that must be addressed to maintain high accuracy as data volume grows.

The proposed project, involving the development of an LSTM autoencoder for online anomaly detection holds a contribution exploring the identified gaps. The employed solution contributes technically by exploring the power of LSTM autoencoder within online anomaly detection problems. A robust solution in real-time steaming data with temporal dependencies. The project provides a valuable tool with the ability to adapt to the evolving data enhancing its suitability and practical utility for distinct domains that require a continuous monitoring and anomaly detection. Moreover, the

project offers a comprehensive solution for online anomaly detection in unsupervised systems. By addressing the challenges of online data scaling and real-time anomaly detection, it contributes to the growing field of streaming analytics.

CHAPTER 3 METHODOLOGY

An overview of the problem faced, including a detailed NDTech 2.0 operating system study, the available data that will be used and analysed and methodologies that will be required and applied to meet the project goals.

This methodology chapter will cover all the required stages to deliver the established objectives, starting with a complete examination of the issue at hand, including an in-depth exploration of NDTech 2.0's operation. Later, will be introduced the available data, supplemented by a revealing Exploratory Data Analysis (EDA) to uncover hidden insights and patterns. Finally, the focus will shift to model development, where all the available resources were leveraged to enhance the LSTM autoencoder's capability to extract latent features from the data. Every component of model architecture development, training, and optimization is carefully explored, culminating in the definition of thresholds necessary for the anomaly detection framework. This chapter sets the methodical approach employed to tackle the challenges at hand, placing the foundation for the subsequent chapters of this study.

3.1 PROBLEM DESCRIPTION

The TCA detection process, as previously highlighted in this report, holds a critical significance for the cork industry where guaranteeing its quality and reliability is crucial. With the prevalence of advanced and complex machinery becoming the new standard in most manufacturing facilities introduces new challenges and constrains in control procedures, calling for the adoption of new and more efficient practices. NDTech stands as a remarkably sophisticated technology, a vivid example of this scenario as the complexity of the multi-component architecture demands a close control to ensure the proper functioning of all the system elements, which current relies on a periodic manual control procedure making it susceptible to human error and limitations on both efficiency and resources demands.

This project is developed following the identification and desire to pursue the possibility to improve the current control implemented on NDTech, taking advantage of the data available in the second generation of the technology, NDTech 2.0, where is recorded the information regarding the different

variables and indicators provided by the sensors that measure distinct functional parameters from the components of the system. In 2022, Rogério Nunes conducted an initial investigation which started with the identification of the relevant features for the intended task, employing both analytical techniques as well as the know-how from individuals engaged in day-to-day system operations. The data was also examined to extract valuable insights and gain a comprehensive understanding of its behaviour. Subsequently, the performance of various anomaly detection algorithms was compared, to establish the best option to address the problem (Nunes, R., 2022).

The aim of this work is to take the previous efforts further and move forward into new unexplored areas and gather more information and knowledge regarding the data as well as explore a different algorithm as a solution for the problem. This first goal involves continuing to study the data, for longer periods of time and more in detail for some specific scenarios to develop and improve current understanding of the available data. The second goal is to test and take advantage of the capacities of machine learning techniques, specifically autoencoders, to develop a real-time automated anomaly detection system that could be implemented on site and used by the NDTech operators. In pursuit of these goals, we have formulated two primary research questions:

1. Can anomalies be detected using autoencoders?

This question searches into the core of this investigation, seeking to determine whether autoencoders, where the correct choice and it can effectively detect anomalies within the NDTech data.

2. Can the current control system be improved with this approach?

This question explores the practical implications of this research. It seeks to conclude whether leveraging autoencoders to develop a real-time automated anomaly detection system could improve the existing control system utilized by NDtech operators.

3.1.1 NDTECH 2.0

This part of the text will delve deeper into examining the structure and operation of the NDTech 2.0 system. This understanding is crucial to provide a solid foundation for the subsequent stages of the project. The NDTech 2.0 chromatography system comprises three steps: sample preparation,

chromatographic separation of the sample components and, finally, TCA quantification on the detector.

Each module of the second version of the of the technology, NDTech 2.0, is composed by four processing units that share the same feeding system that provides the cork stoppers to be analysed. The robotic transport line decides which unit need to be supplied and place the cork accordingly. The structure of each NDTech 2.0 Punit is illustrated in the *Figure 3.1*.

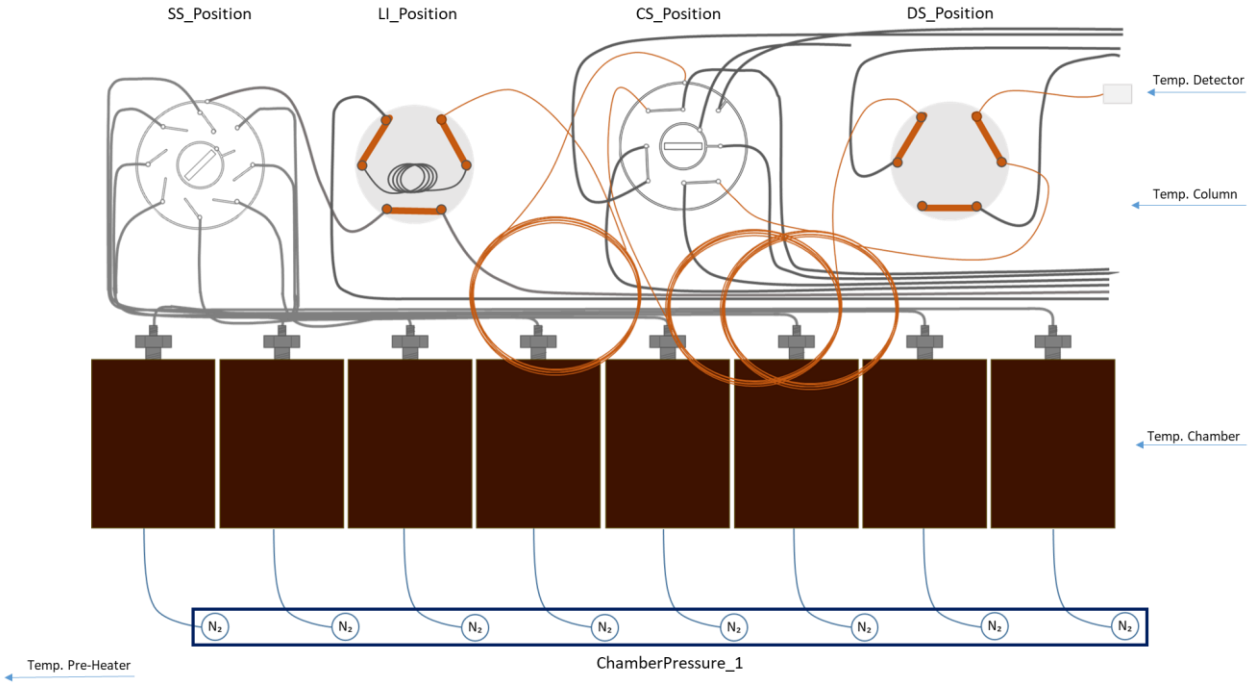


Figure 3.1 – General layout of a processing unit within NDTech 2.0 system

The cork stoppers are inserted on the pre-heater (variable *Preheater_Temp*) where they are heated beforehand, until a carriage, in control of all transportations during the entire process, allocate it into an available incubator chamber. The chambers temperature (variable *Chamber_Temp*) is controlled and held at a high value to prepare the cork for the chemical analysis. The first valve, Stream Selector (SS) (variables *SS_Position* and *SS_Torque*), is responsible to decide which of the 8 chambers is being under analysis at any time, by allocating the underlying column in place to carry the gas sample forward, while the remaining are being cleaned with a continuous nitrogen stream. The next vale, Load Inject (LI) (variables *LI_Position* and *Li_Torque*) by using a loop, ensures that the volume and flow of the sample to be injected in the pre column is controlled and is similar for all chambers, which

may have flow variations for various reasons. This process is possible with a connection with one of the nitrogen injectors, the “Carrier” (variable *Box104)*, to facilitate the coordination and timings to control the sample destination.

At this stage, the Column Selector (CS) (variables *CS_Position* and *CS_Torque*) valve, is responsible to select the analytical column to which the sample will be transferred. Moreover, a macro separation of the components that make up the sample coming from the pre-column is carried out. When the valve receives the sample from the pre-column, it eliminates and separates part of the sample that is not in the range where the TCA is expected to be due to the properties of the column that alter the arrival time of the different chemical compounds in the sample, in order to isolate and discard the unwanted compounds to improve the chromatograph's analysis (heart cutting). In the column, a micro separation and isolation of certain compounds from the sample happens. This component has the most variables associated with it, as it has 4 different nitrogen injectors, two of which are responsible for injecting nitrogen into the analytical columns (*Box105, 107*) and the other two which inject nitrogen into the purges (*Box106, 108*).

Lastly, the Detector Selector (DS) (controlled by variables DS_Position and DS_Torque) establishes the connection between the analytical columns and the detector. It toggles between the columns in an alternating fashion, precisely timing the transfer of the targeted sample portion to the detector while discarding the undesired portion.

3.2 THE DATA

The NDTech 2.0 system's data captures various variables and indicators originating from sensors responsible for gauging diverse functional parameters within each system component. This data is store and can be accessed through an SQL server database. The data is composed with a total of 87 different variables related to the operation of the machinery and its complex environment. Not all the features may be considered as useful and crucial to the purpose of this project, as some of them may not be directly related to the description and detailing of the operation and in particular to the anomaly detection problem. In the preceding study conducted by Rogério Nunes, a comprehensive set of 57 variables was identified as pivotal for addressing the specific issue at hand (*Figure 3.2*). These variables were deemed capable of effectively encapsulating the operational aspects of the machinery (Nunes, R., 2022).

Box104_EffX_Min	Box104_EffX_Avg	Box104_EffX_Max	Box104_OutY_Min	Box104_OutY_Avg	Box104_OutY_Max
Box105_EffX_Min	Box105_EffX_Avg	Box105_EffX_Max	Box105_OutY_Min	Box105_OutY_Avg	Box105_OutY_Max
Box106_EffX_Min	Box106_EffX_Avg	Box106_EffX_Max	Box106_OutY_Min	Box106_OutY_Avg	Box106_OutY_Max
Box107_EffX_Min	Box107_EffX_Avg	Box107_EffX_Max	Box107_OutY_Min	Box107_OutY_Avg	Box107_OutY_Max
Box108_EffX_Min	Box108_EffX_Avg	Box108_EffX_Max	Box108_OutY_Min	Box108_OutY_Avg	Box108_OutY_Max
ChamberPressure_1_Min	ChamberPressure_1_Avg	ChamberPressure_1_Max	ChamberPressure_2_Min	ChamberPressure_2_Avg	ChamberPressure_2_Max
CS_Torque_Min	CS_Torque_Avg	CS_Torque_Max	DS_Torque_Min	DS_Torque_Avg	DS_Torque_Max
SS_Torque_Min	SS_Torque_Avg	SS_Torque_Max	LI_Torque_Min	LI_Torque_Avg	LI_Torque_Max
CS_Position	DS_Position	SS_Position	LI_Position	AnalTime	
Chamber_Temp	Detector_Temp	Column_Temp	Preheater_Temp		

Figure 3.2 – List of NDTech 2.0 variables

The systems produce in each individual cork stopper analysis cycle, of about 20 seconds, a total of 200 registrations in parcelled intervals of 100ms, which leads to a huge amount of data available. Due to the storage cost and computational power demands, the manufacture choose to limit the registered instances to a single observation for each cork analysed. In order to achieve this, was set the mean value of the intermediate measurements during the 20 seconds cycle for the continuous variables measuring components temperatures and valve positions. For the remaining features, besides the mean value is also registered the minimum and maximum values obtained for each cork, during each cycle.

In addition to the previously emphasized 57 variables, it is pertinent to note the existence of three other features associated with the TCA analysis itself. These features encompass *PeakValue*, quantifying the peak value of TCA detected in each cork, *Peaktime*, measuring the analysis moment when the peak compound was identified; and *DeltaTime*, which gauges the duration of the entire chromatography analysis process. Once the main objective of this work is not centred on TCA quantification, the former feature, PeakValue, holds no value. This is justified by the inability to directly associated the detected value of the compound with a faulty operation of the system. However, for the 2 remaining variables, cannot be drawn the same conclusions promptly, as they may carry some valuable information for the anomaly detection algorithm and potentially offer relevant insights to identifying anomalies within the system. Consequently, these features will also be incorporated into following analyses to establish whether is justified their integration in addressing the problem effectively.

The NDTech system lacks a classification system with the ability to label the available data in real-time independently of being obtained manually or in automated mode. It cannot identify an observation as normal or abnormal within the regular functioning of the technology. This specific detail holds significant importance, influencing the viewpoint and methodology required for developing the anomaly detection algorithm.

3.2.1 EXPLORATORY DATA ANALYSIS

The data interpretation marks the initial step towards comprehending the intrinsic traits of the problem at hand. By unveiling insights hidden within each dataset feature, identifying trends or patterns, ultimately laying a foundation for subsequent analysis and strategies to address the anomaly detection challenge effectively. In this case, is not only confirmation of the prior findings undertaken by Rogério Nunes, but also an attempt to push boundaries, harnessing new and novel insights offered by the data.

For the data exploration phase, was selected a dataset that spans from March 15, 2023, 12:00h, to March 17, 2023, 12:00h time interval encompassing 2 working days of data which contains a total of 34,265 observations over the defined period.

3.2.1.1 DISTRIBUTION

In the initial phase of this Exploratory Data Analysis (EDA), the focus lies on understanding the data's representativeness by employing a variety of visualizations tools such as box plots, violin plots, and scatter plots that can help achieving the desired objectives. These primary goals are centred around gaining insights regarding the distribution of the data, the difference between the features available, identify possibly subgroups within the data, identify potential outliers and recognise possible patterns and relationships amid variables.

Due to the diverse nature of measurements represented by various sets of features, it was anticipated differences in how the data is characterized and distributed. For instance, when examining the data points related to variables tracking valve positions, it's evident that they exhibit oscillations around specific fixed valve rotation points, as illustrated in *Figure 3.3*. It's worth noting that this trend doesn't

apply to the Stream Selector feature, which is not scaled within the same range [-360º, 360º] as the other three. Very distinct are the representations of the temperatures recorded across the different components, which appear to be more stable over time, with minor fluctuations occurring around the established setpoint.

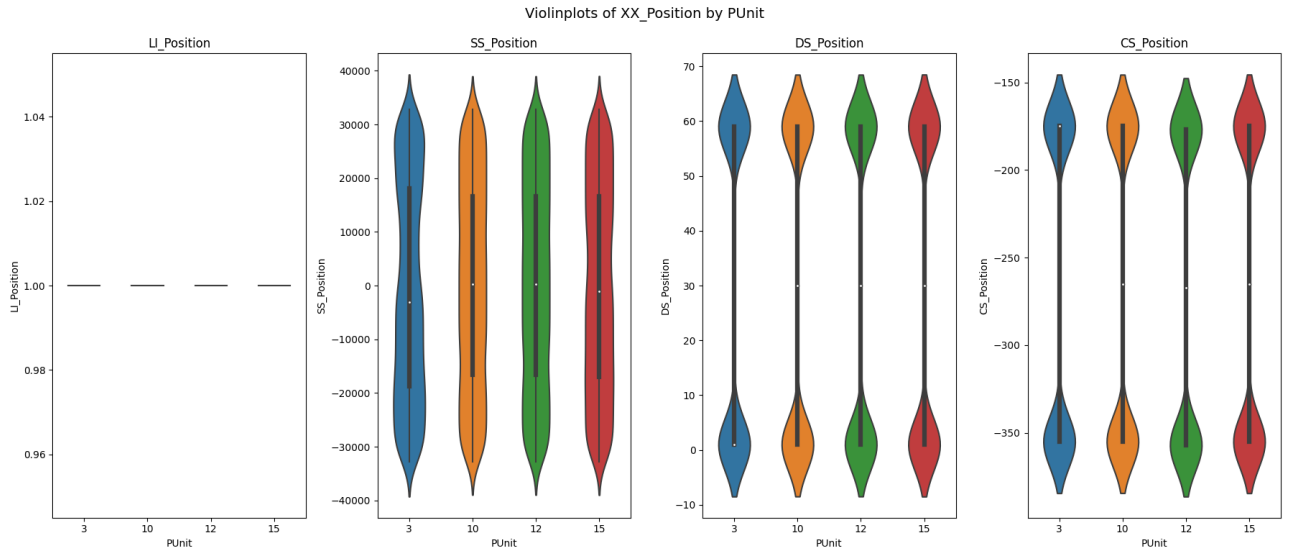


Figure 3.3 – Violinplots of position variables for the distinct valves (LI, SS, DS, CS) by processing unit

Regarding the operation behind the gas circulation that allows the correct synchronisation of the sample and the timings of the chromatographic analyses recorded by the variables labelled as *Box (Box104, Box105, Box106, Box 107, Box108)*, there is a noticeable divergence in the range of the values registered, based on the specific function allocated to each feature. By other words, it's perceivable that both Box106 and Box108 variables, which are associated with the gas purge operation of the Column Selector, exhibit a similar range of values, distinct from the "Mini-carriers" (*Box105* and *Box107*), or the features responsible for injecting nitrogen into the same valve. Finally, a distinctive behaviour from all previous mentioned features, can be spotted for the flow measurements for the injection of the carrier gas into the Load Injector (*Box104*). Therefore, it can be inferred that there is a subdivision within the features controlling the pneumatic transport systems, based on their respective assignments (as depicted in *Figure 3.4)*, both in terms of the gain component (EffX) and the opening/closing of the controller to maintain the pressure setpoint (OutY).

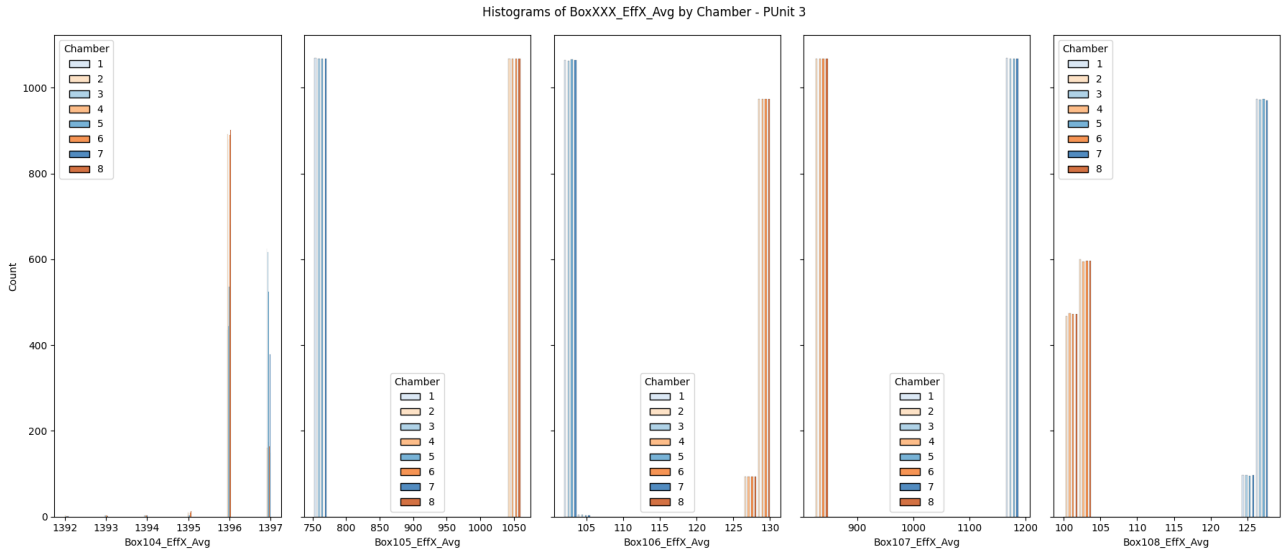


Figure 3.4 – Histograms of the average range of values for the Box feature *(Box104, Box105, Box106, Box 107, Box108)* by chamber for Punit 3

During the data exploration phase, it was monitored that minimal variations appeared in the distribution of all three statistical measures related to the ChamperPressure_2 feature, which oscillated between 0 and -3. Briefly after reported this situation and subsequently discussed with both NDTech collaborators and manufacturers, it was concluded that although the variable was configured and prepared for use, it had not been allocated to any specific component, carrying no value for this project and therefore it was discarded for all the future analysis.

Despite the intrinsic diversity of the variables, it is the clear distinction between the range of operations across the various process units, which is reasonable due to their independent and individual function. However, one particular finding, previously acknowledged by Rogério Nunes, is the significant discrepancy in the operating ranges between the cork stoppers analysed in the even and odd chambers within each of the PUnits. This scenario is predominant in most of the features, with some exhibiting a substantial variance in the range of values for the two subgroups (as depicted in *Figure 3.5*) while other revealing more subtle adjustments. The observed variations in the operational characteristics between the odd and even chambers can be attributed to the fact that each group relies on a distinct analytical column to connect them to the detector.

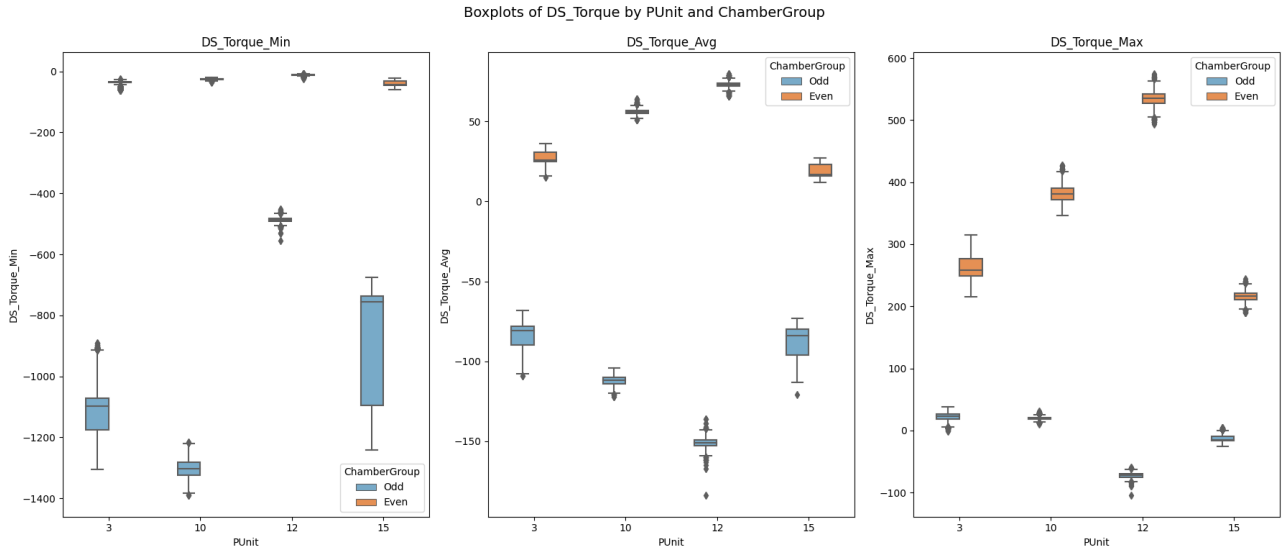


Figure 3.5 – Boxplots of the variable DS_Torque by chamber group

3.2.1.2 MISSING VALUES

The presence of missing values in the dataset can compromise the and impact the reliability of subsequent analyses. Consequently, detecting their existence, investigating the root cause, and forming appropriate strategies to address them is crucial. Even though no cork stoppers analysed by NDTech were found to contain missing values in the selected dataset for this analysis, later examinations revealed the presence of *Null* values that affected the required analysis for the purpose of this work.

To address this issue, a more extensive time interval was inspected to identify and study the presence of missing values on the dataset. Over the course of a month of collected data a total of 114 observations containing absent data. Among these observations, 107 exhibited missing values exclusively in the *SS_Torque_Min* feature. Notably, the values register for the maximum (*SS_Torque_Max)* and average (*SS_Torque_Avg*) variables also raise attention as they vary from the values for all remaining instances without the presence of any null values (*Table 3-1*). In the remaining 7 instances, 4 had no recorded values for the *SS_Position* feature and the remaining 3 had a total of 56 variables with no measurements. Despite the relatively limited scale of the discovered situation its factual implications on the main task of this project could not be dismissed.

Table 3-1 – Table with examples of observations with missing values in the *SS_Torque_Min* variable

Analtime	*PUnit*	*Chamber*	*(…)*	*SS_Torque_Min*	*SS_Torque_Avg*	*SS_Torque_Max*	*(…)*
2023-04-03 10:05:49.242	12	1	…	NaN	9999.0	-32768.0	…
2023-04-03 10:06:09.236	12	2	…	NaN	9999.0	-32768.0	…
2023-04-03 10:06:29.239	12	3	…	NaN	9999.0	-32768.0	…
…	…	…	…	…	…	…	…
2023-04-03 12:37:09.362	12	5	…	NaN	9999.0	-32768.0	…

After some examination it was not possible to detect the source of this non desired instances, they could arise either from technical issues, data entry problems or different factors. The 107 detected observations with Null values in the *SS_Torque_Min* variable, occurred all in the same PUnit for consecutively corks analysed over all chambers, during a total of 35 minutes of processed instances in that process unit. The remaining 7 cork stoppers were isolated registrations during the period analysed. In either of these cases was possible to establish a connection with a faulty operation of the system.

The findings of this research, while no conclusions were drawn regarding the underlying reasons for its presence, required some data treatment to allow any subsequent analysis. In this case it was decided to, for the observation with no values registered for the *SS_Torque_Min*, that cover the vast majority of the detected situations, to replace both the missing values of this variable but also the misleading values for *SS_Torque_Max* and *SS_Torque_Avg*, by the mean value to reduce the impact of these deviations on this work's main goal. For the cases where the missed values were detected on the position of the stream selector valve, the missing values were replaced by the mode value of the variable, while for the remaining where a significant portion of the variables had no information these observations were excluded from future analysis.

3.2.1.3 OUTLIERS

Outliers, which are data points that deviate from the expected behaviour and remaining data distribution, often possess the capacity to influence and distort statistical analyses, machine learning

models or decision-making processes. These data points are often associated with anomalies or faulty processes and their presence can distort data assumptions and lead to misleading interpretations. Therefore, a comprehensive understanding of the presence of outliers is indispensable for any data to derive accurate and reliable insights from their dataset.

In this segment will be analysed the presence of outliers in the selected data both using the interquartile range (IQR) method as well as the Z-score transformation technique. From the previous findings of data groups with similar distributions based on the chamber group of each PUnit, the following tasks will be developed individually for each of the subgroups determined.

Interquartile range – IQR

As far as the first method is concerned, developed by Tukey et al., in 1977, it focuses on the spread of data withing the middle 50% of values specifically the range between the first (Q1) and third quartile (Q3), that is named as the interquartile range (IQR $=$ Q3 – Q1). Any point that falls below $Q1 - 1.5 * IQR$ or above $Q3 + 1.5 * IQR$ is considered as a potential outlier.

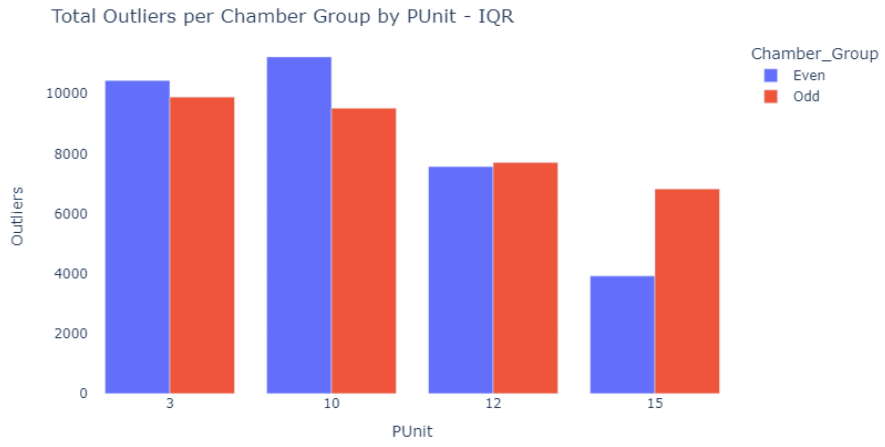


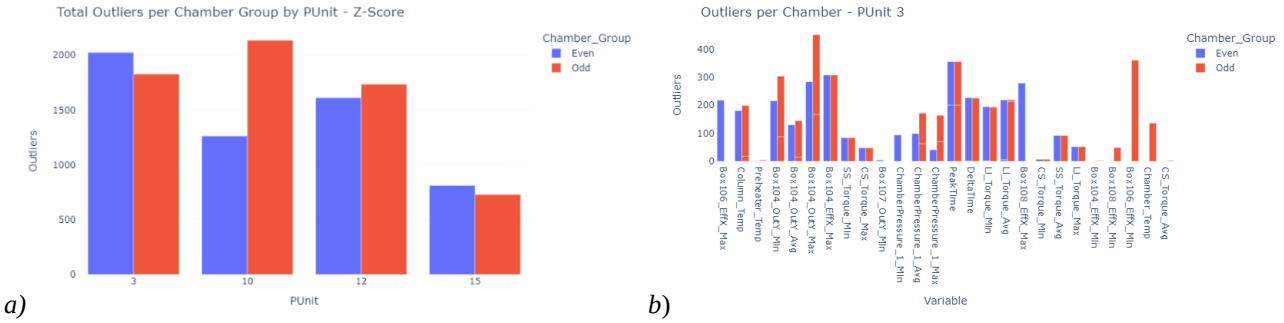
Figure 3.6 – Bar plot with the total number of identified outliers with IQR method for each chamber group within each PUnit

When assessing the presence of outliers in the data using the interquartile method, a crucial observation is that in many features, the IQR value is 0. This means that any value distinct from the mode is considered as an outlier, regardless of how distant it actually is from the central tendency. This situation has resulted in a potential overestimation of the number of outliers, leading to a high number of identified outliers, as illustrated in *Figure 3.6.*, where nearly half of them falling into this

category (IQR = 0). These findings have raised concerns about potentially inflated outlier records that do not accurately represent true outliers. As a result, was employed the Z-Score method to mitigate the issue of dealing with misleading statistics.

Z-Score

The Z-score method introduced by Iglewicz and Hoaglin in 1993, is a statistical technique where the outliers are identified by measuring how many standard deviations a data point is away from the mean. The Z-score for each instance is calculated by subtracting the mean and dividing by the standard deviation ($\text{Z-score} = \frac{X-\mu}{\sigma}$). Any data point with an absolute Z-score value above 3 is considered as a potential outlier.



a) *b)*

Figure 3.7 – a) Bar plot with the total number of identified outliers with Z-Score method for each chamber within each Punit; b) Bar plot with the distribution of the outliers from PUnit 3 for each chamber group across the variables

The total count of data points exceeding the threshold is notably reduced compared to the previous method (*Figure 3.7 a*). Nevertheless, the presence of outliers is still meaningful. In *Figure 3.7 b*, as depicted in the right-hand plot, we observe that the outliers within the data originating from processing unit 3 are dispersed across multiple features, which are consistent with those found in the remaining data.

3.2.1.4 DATA SCALLING

The complexity of NDTech technology leads to a wide diversity of measurement units from the retrieve data, with very different range of values, which can mislead the training process interpretation of that information by the algorithm leading to certain features dominating the learning process. Data scaling is therefore a crucial step working with neural networks in general, and in this case particularly with autoencoders. Finding a common scale for all the features, can lead to a faster convergence, more stable training, and better generalization of autoencoder models, or in other words scaling the data can help ensuring the learning of meaningful representations from the input data by more efficiently capture underlying patterns, resulting in improved model performance and better results on various tasks.

For this project it was adapted the *Z-score* technique with two different variations based on the specific task demands, the first one, the traditional method where all the data is known, so both the training set as the test set can be scale, based on the mean and standard deviation values calculated from the full datasets. This ensures the data homogeneity, helping to enhance the model performance and analysis. Although, for the online approach it's not possible to apply the same exact method, as the new data arriving is unknown and can shift the environment dynamics, raising new challenges that required a solution that can't be addressed by the traditional scaling technique. To the streaming application, the *Z-score* values were estimated based on updating approximated values of the mean and standard deviation, that will be detailed later in this report.

3.2.1.4.1 ONLINE DATA SCALLING

As has been mentioned before the data scaling technique used previously was not suitable for an online system, where would be required to store the entire information regarding the new data to calculate the mean and standard deviation to compute the Z-score value for each observation. The computational power and time required to aggregate calculations estimated throughout the execution is unfeasible.

Therefore, for landmark window approaches is possible to incrementally compute the Z-score value without having to store all the data points scanned until a certain point, instead only 3 numbers are required to achieve. By storing *n*, number of instances analysed until time *t*, the sum of all data point

until t, $\sum(x)$, and $\sum(x^2)$, the sum of the squares of the data points. With this information is possible to compute the necessary statistical information to calculate the Z-score, where

$$\mu_t = \frac{\sum(x)}{n} \text{ and } \sigma_t = \sqrt{\frac{\sum(x^2) - \left(\frac{\sum(x^2)}{n}\right)}{n-1}}$$ (Gomes et al., 2019).

3.2.1.5 CORRELATION

An essential aspect of this data exploration is the investigation of the relationship between variables, where potential connections and patterns within the data can be revealed. To support this task, the correlation analysis was applied to assess the strength and direction of linear relationships between pairs of features. The target of this analysis was not only to recognize whether changes in one variable were associated with changes in another variable but also to quantify the strength of these relationships as manifested in the data. Ultimately, the purpose of this analysis was to understand whether the discovered findings with established patterns or expectations based on the prior knowledge of to the variables and operation of the NDTech.

Once the conditions of assuming a normal distribution of the variables as well as its linearity and homoscedasticity were not established, both *Spearman* and *Pearson* methods proved unsuitable for the dataset. Consequently, the correlation was calculated using the *Kendall* correlation coefficient, which does not rely on the assumption of normality and its less sensitive to the presence of outliers.

The correlogram (see *Figure 3.8*), were generated after excluding variables with a very low number of unique values. The obtained correlation values demonstrate that, for the majority of pairs of features, there is little to no correlation between them. Nevertheless, there are some exceptions, such as in the Punit 15 for the variables *Box106_EffX_Max* and *Box108_EffX_Max* with a correlation index of higher than 0.9 indicating a strong positive linear relationship between them.

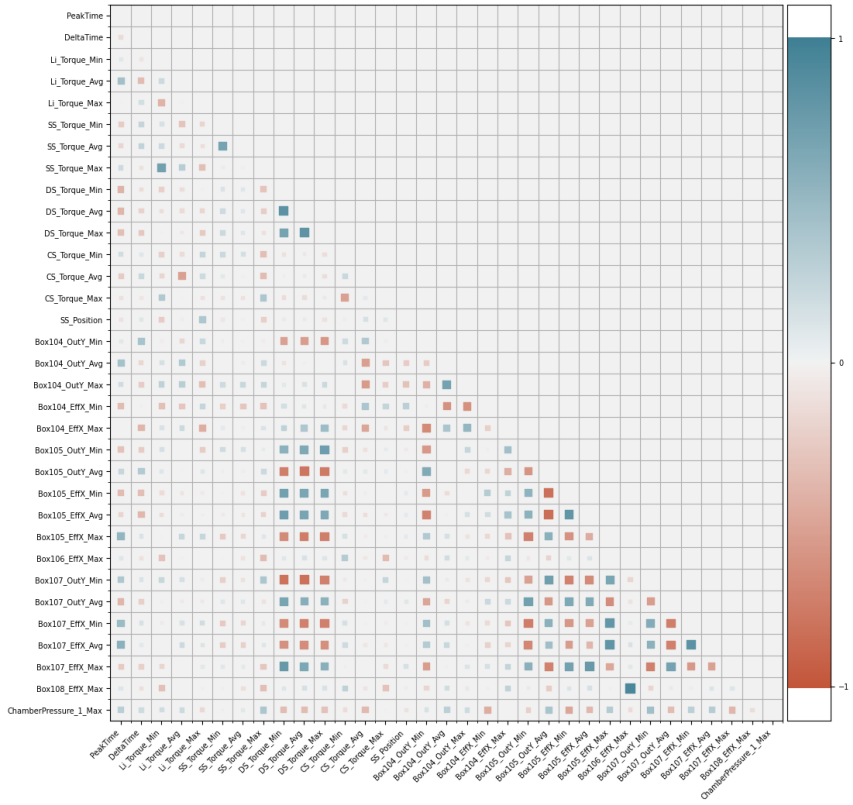


Figure 3.8 – Correlogram for PUnit 15

Even though the correlation analysis didn't reveal very strong relationships between variables, it did provide an indication of some degree of association among certain sets of features. For instances, from the correlogram above, the correlation between *Box105* & *Box107*, may be explained by the connection by the interconnected nature of these variables and their roles within specific processes of the NDTech operating system. Additionally, correlations between different statistical measures of a variable are reasonable, given the specific structure of how values are recorded and stored for each feature.

3.2.1.6 KERNEL PCA

The Kernel Principal Component Analysis (KPCA), introduced in 2002 (Shölkopf et al., 2002) is an extension of the classical PCA algorithm, capable of better capture complex and nonlinear features from the data by using a kernel function to perform PCA in a high-dimensional space which is non-

linearly related to the input space. Traditional applications of KPCA revolve around dimensionality reduction, where the maximum variance within the data is captured into a lower-dimensional representation, but in this case, it will be used as a visualization tool (Nasser et al., 2006) with the intention to evaluate and confirm the previous findings of possible subgroups inside the data.

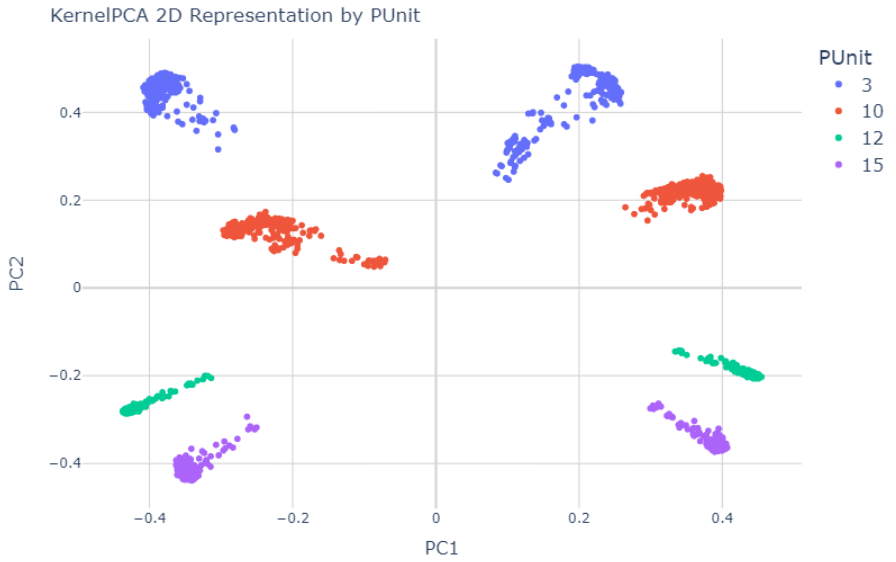


Figure 3.9 – Kernel PCA 2D representation by PUnit

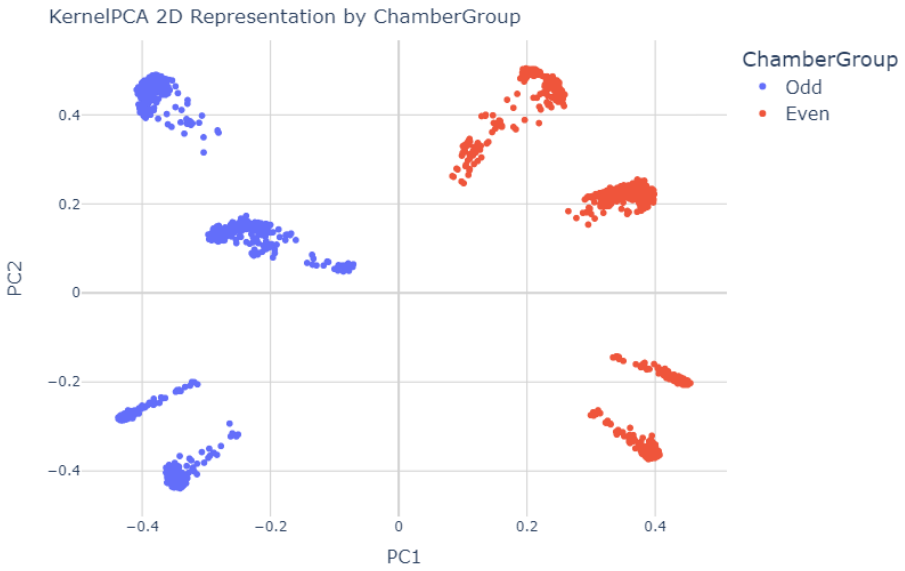


Figure 3.10 - Kernel PCA 2D representation by chamber group

When observed the 2-dimensional scatter plot by the KPCA two main components labelled by the corresponding PUnit (*Figure 3.9*), it is straightforward the difference between the data points representing each processing unit, supporting the previous assumptions regarding the distinctive function and underlying data distribution between them. Besides the discrimination from PUnits it is noticeable the presence of 2 distinctive clusters inside each PUnit's data points, which with a more detailed investigation were proved to be related to the chamber where the instances were analysed, more precisely the group of chambers. From *Figure 3.10*, is clear the difference between the data points analysed in even chambers and odd chambers. This difference can be justified by the shared analytical columns shared by the chambers in each group.

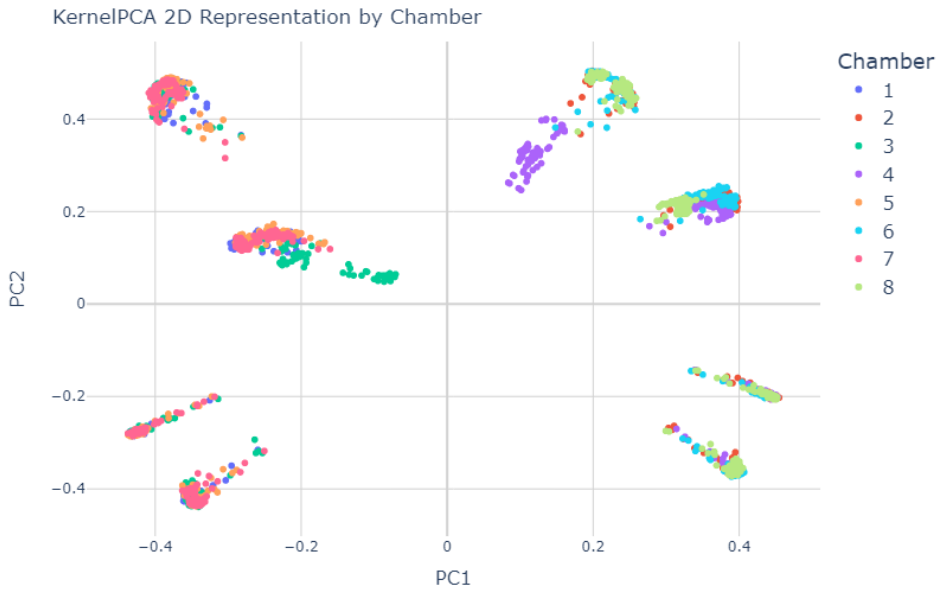


Figure 3.11 - Kernel PCA 2D representation by chamber from all the PUnits

Finally, to check whether there were significant differences among the data points analysed by the different chambers inside the each (*PUnit – ChamberGroup)* subgroup was examined the Kernel PCA representation individually coloured by the chamber, where its visible that there isn't a significant differentiation among the data points from the clusters previously identified (*Figure 3.11*). Therefore, this information is crucial for the upcoming task as the data implies that each cluster of even and odd chambers from each processing unit should be analysed individually due to the substantially different function evidenced.

3.3 THE MODEL

The current number of available and applicable algorithms for anomaly detection problems is extensive, but the right fit between the selected method and the system peculiarities can leverage the output obtained and increase its performance. For this particular challenge, there were the intent to explore the capacities of machine learning techniques, namely autoencoders, due to their unsupervised learning technique to learn the data but also their applicability to online problems.

3.3.1 LSTM AUTOENCODER

For the algorithm development, the Keras library in Python was employed to build an efficient autoencoder model tailored for anomaly detection. The model featured a symmetric architecture encompassing an encoder and a decoder. The number of layers and the dimensions of each layer were chosen to reach a balance between model complexity and the need to avoid overfitting.

Whitin this architecture, both the encoder and decoder components formed of 2 fully connected Long Short-Term Memory (LSTM) layers with dimensions of 24 and 4 units. Both the encoder and decoder layers used hyperbolic tangent activation function, or often abbreviated as "tanh", which is a scaled and shifted version of the hyperbolic tangent function. It is a mathematical function that maps input values to a range [-1,1]. Essentially, a timestep of 4 was employed, representing a complete cycle of analysis within a NDTech 2.0 module. This specific choice of timestep corresponded to a holistic view of the data, encapsulating the insights derived from 32 cork stoppers analysed in each cycle by all processing units.

A TimeDistributed (Dense) layer was introduced within the decoder architecture to facilitate the restoration of the data to its original shape. This layer iterates over the temporal dimension of the encoded data, applying a fully connected Dense transformation at each time step allowing the decoder to unfold the compressed representation back into the initial temporal sequence, ultimately restoring the data to its original form.

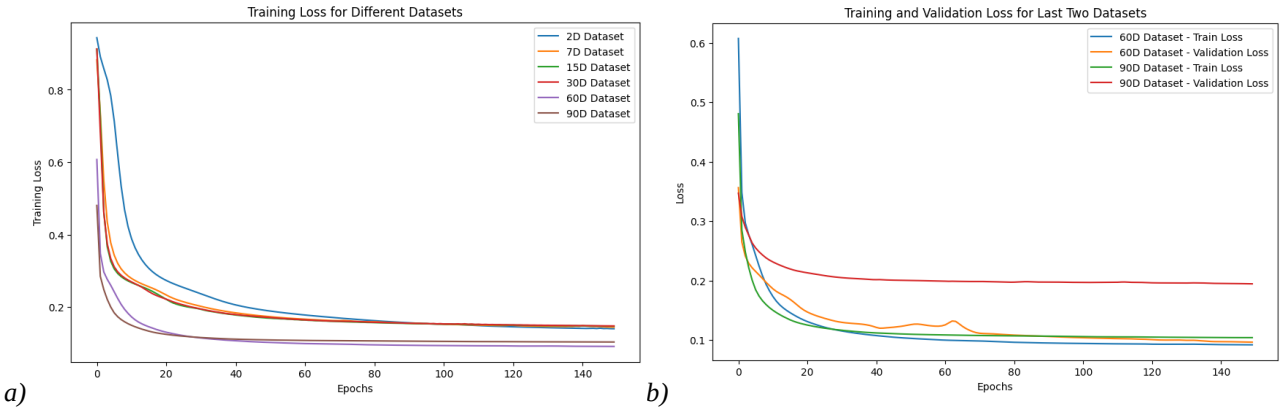
The Mean Absolute Error (MAE) were employed to guide the training of the model. The loss function quantifies the divergence between the reconstructed output from the original output. By minimizing this divergence throughout the training iterations, the autoencoder is incentivized to learn the intricate

data patterns in order to accurately recreate the input instances, usually normal observations increasing the anomaly detection capabilities.

3.3.1.1 ALGORITHM TRAINING

The unsupervised environment of NDTech's data does not facilitate the best and optimal training scenario for the algorithm, where the training dataset would be composed of only normal observations. Instead, the dataset would be used in its original state, where the presence of potential anomalies in the training dataset is expected, although in a reduced number. The only exclusion from the training dataset would be the periods under the malfunction's simulations, intentionally caused to be studied.

In the absence of labelled data, the prime objective of the algorithm training and optimization shifts from the traditional focus on accuracy-focused paradigms to concentrating on narrowing the reconstruction error of the original inputs. This emphasis on minimizing reconstruction error aims to enhance the algorithm's ability to meticulously recreate instances. Nevertheless, it's crucial to acknowledge that this quest, particularly in the presence of potential abnormal instances in the training dataset even though in limited proportions, can reach a point where it might inadvertently obscure the algorithm's primary objective of distinguishing between the two observation types. This potential challenge is underscored by the reduced representativeness of abnormal instances in the training dataset.



a) *b)*

Figure 3.12 – a) Plot with models training loss with the different length datasets for the odd chambers of PUnit 10; b) Plot with models training and validation loss for the 60 and 90-days datasets for the odd chambers of PUnit 10

Concerning dataset size utilized for training the AD algorithms experimentation with various time spans, ranging from 2 days to 90 days, was undertaken. Notably, the algorithms demonstrated enhanced learning with larger datasets, as evident in *Figure 3.12 a*. However, during these investigations, the 60 days dataset were the optimal performer. When verifying the validation loss across 60 and 90-day intervals, revealed that the 60-day dataset reached lower loss values, as can be seen in the second graph on *Figure 3.12 b*, implying that the model's performance was better, and it was able to make more accurate predictions on the validation data when trained on the 60-day dataset. This outcome held true for each subgroup, leading to the choice of training all the datasets with the 60-day dataset.

The training process was facilitated by a batch size of 72, spanning 150 epochs, and employing the 'Adam' optimizer. Additionally, to enhance training efficiency, an early stopping function was implemented. This function monitored the loss value during training and, if the loss did not decrease for three consecutive iterations, training procedure were terminated.

3.3.1.2 ALGORITHM OPTIMIZATION

The algorithms optimization was implemented using the Keras Tuner library available on python, which is a versatile tool that efficiently navigates through the possible combinations to identify optimal configurations tailored to each subgroup's unique characteristics. The idea was to extract the best possible models, fine-tuned to enhance performance of each model considering a comprehensive range of parameters, culminating in a comprehensive exploration of potential model architectures.

In the following table (*Table 3-2*) are listed all the possible combinations that were tested to find the best model possible:

Table 3-2 – Table with the range for all parameters considered during the optimization

PARAMETER	RANGE
LSTM Layers	[1,2,3]
Layers Units	[6, 24, 48]
Regularizer L1	[0, 0.1, 0.01]
Regularizer L2	[0, 0.1, 0.01]
Dropout	[0, 0.1]
BatchNormalization	[True, False]
Learning Rate	$[1e^{-5}: 1e^{-2}]$, *log* sampling

The exploration covered various parameters, with the integration of regularization techniques that play a role in controlling the network complexity and preventing overfitting. The L1 and L2 regularization techniques add penalty term to the loss function during training helping prevent overfitting by discouraging the model from fitting noise in the data and encouraging it to focus on the most important features. L1 regularization (Lasso), encourage sparsity and feature selection, buy imposing a proportional penalty to the absolute magnitude of the model's weights. On the other hand, L2 regularization, referred to as Ridge regularization, penalizes the squared magnitude of weights, promoting a more balanced distribution.

Alternatively, the dropout enhances the generalization to unseen data from the model, by intentionally adding noise to the data. Dropout randomly drops out a fraction of neurons by setting them to 0 during each training iteration. This forces the network to learn a more robust representation of the data, by learning redundant representations and prevents it from relying too heavily on specific neurons, failing when applied to new and different data (Srivastava et al. 2014).

Batch normalization is a technique used to stabilize and accelerate the training of deep neural networks. It involves normalizing the inputs to each layer within a mini-batch of data. This stabilization of the inputs, can decrease the changes on the distribution of inputs to a layer leading to a faster convergence and therefore accelerating the training process (Ioffe et al., 2015)

Upon identifying the optimal combinations for each subgroup, the subsequent phase involved training the models with the optimal configurations and saving them to latter be applied when required.

3.3.1.3 THRESHOLD

The inherent variability and unpredictability in the dynamic scenario were not suitable for a static and constant threshold throughout the analysis. The characteristic of the faced problem required a dynamic and evolving threshold strategy capable of moulding to the data patterns and shifts. With the real time system perspective, the threshold was defined based on the recent analysed observations, particularly the 128 last seen data points, according to the following formula:

$$T = (1 + k) * \mu_{128}$$

By estimating the threshold based on a scaling factor of the recent mean value there is a filtering of data points that deviate from the recent historic obtained reconstruction errors, but also capable of capture and adapt to shifts in the data distribution by prioritizing the most recent data, where historical observations lose relevance over time.

3.3.1.3.1 ONLINE THRESHOLD

The threshold estimation while straightforward when all data is accessible, in an online system need some adjustments to prevent the necessity to storing the entire dataset. Like the data scaling approach detailed earlier, it's possible to efficiently improve the methodology requiring less information storage to enhance performance and suitability to deal with streaming data and real-time monitoring. Instead of a landmark window, in this case it involves maintaining a fixed-size window where new instances are added, and old ones are removed incrementally.

In this case was used a technique called Welford's method or Welford's online algorithm introduced by B. P. Welford in 1962 to calculate the mean and variance with low computational and memory requirements. This method involves updating the mean and squared differences from the mean and is a foundational approach for efficiently handling statistical calculations in a sliding window scenario. For the proposed solution was used to update the estimated mean (μ) and squared differences from

the mean ($M2$) in two scenarios, the addition of a new observation and the removal of an old one as follows:

Adding an Observation:

$$Update\ Mean\ (\mu):$$

$$NewMean\ =\ OldMean\ +\ (x\ -\ OldMean)\ /\ n$$

$$Update\ M2\ (M2):$$

$$NewM2\ =\ OldM2\ +\ (x\ -\ OldMean)\ *\ (x\ -\ NewMean)$$

Removing an Observation:

$$Update\ Mean\ (\mu):$$

$$NewMean\ =\ OldMean\ -\ (y\ -\ OldMean)\ /\ n$$

$$Update\ M2\ (M2):$$

$$NewM2\ =\ OldM2\ -\ (y\ -\ OldMean)\ *\ (y\ -\ NewMean)$$

Where:

- *OldMean* is the mean before adding or removing the observation.
- *OldM2* is the *M2* value before adding or removing the observation.
- *NewMean* is the updated mean after adding or removing the observation.
- *NewM2* is the updated *M2* value after adding or removing the observation.
- *x* is the value of the added observation.
- *y* is the value of the removed observation.
- *n* is the current number of observations in the window.

With the updated values, the estimated standard deviation can be calculated based on the M2 value with the following formula:

$$StdDev = \sqrt{\frac{M2}{(n-1)}}$$

In this case only the mean value is required but will be used the Welford's method to calculate it.

CHAPTER 4 EXPERIMENTAL RESULTS

A comprehensive overview of the anomaly detection challenge, encompassing the approach developed to analyse the data, as well as an in-depth malfunction analysis with simulations and implementation of the online approach.

As mentioned earlier, taking into consideration the unavailability of labelled data it does not only makes it impossible to truly measure the performance of the selected model but also hamper its optimization process to better adjust to the data and problem properties. In this chapter will be discussed the architecture's system developed both for an initial exploration phase and the required adjustments for the online approach.

A detailed analysis of the intentionally induced anomalies under a controlled environment will be undertaken to get a better understanding of the reflections on the underlying data from the conditioned situations. The simulations will also be used to evaluate the model performance and reactions to the induced scenarios as a benchmark assistant. Afterwards, the online approach will be tested, and its results will be explored in order to obtain a general perspective of the real experimental operation of the system. Finally, a visualization tool concept will be introduced to illustrate the potential benefits to support the NDTech team on-site monitoring the technology operation with real-time information.

4.1 SYSTEM ARCHITECTURE

The approach developed to create the anomaly detection algorithm to NDTech 2.0 data is summarized in *Figure 4.1*, with 3 main groups. The first involves the transmission and storage of the machinery information into the SQL Server. The second focuses on the development and training of the autoencoders. Finally, the third group is where the data is processed, and the results of the analysis are obtained. The goal was to develop a system capable of monitoring the entire operation of a NDTech module. Although individual analyses were conducted for each identified subgroup, the ultimate purpose was to build an integrated system that could handle all the analyses together.

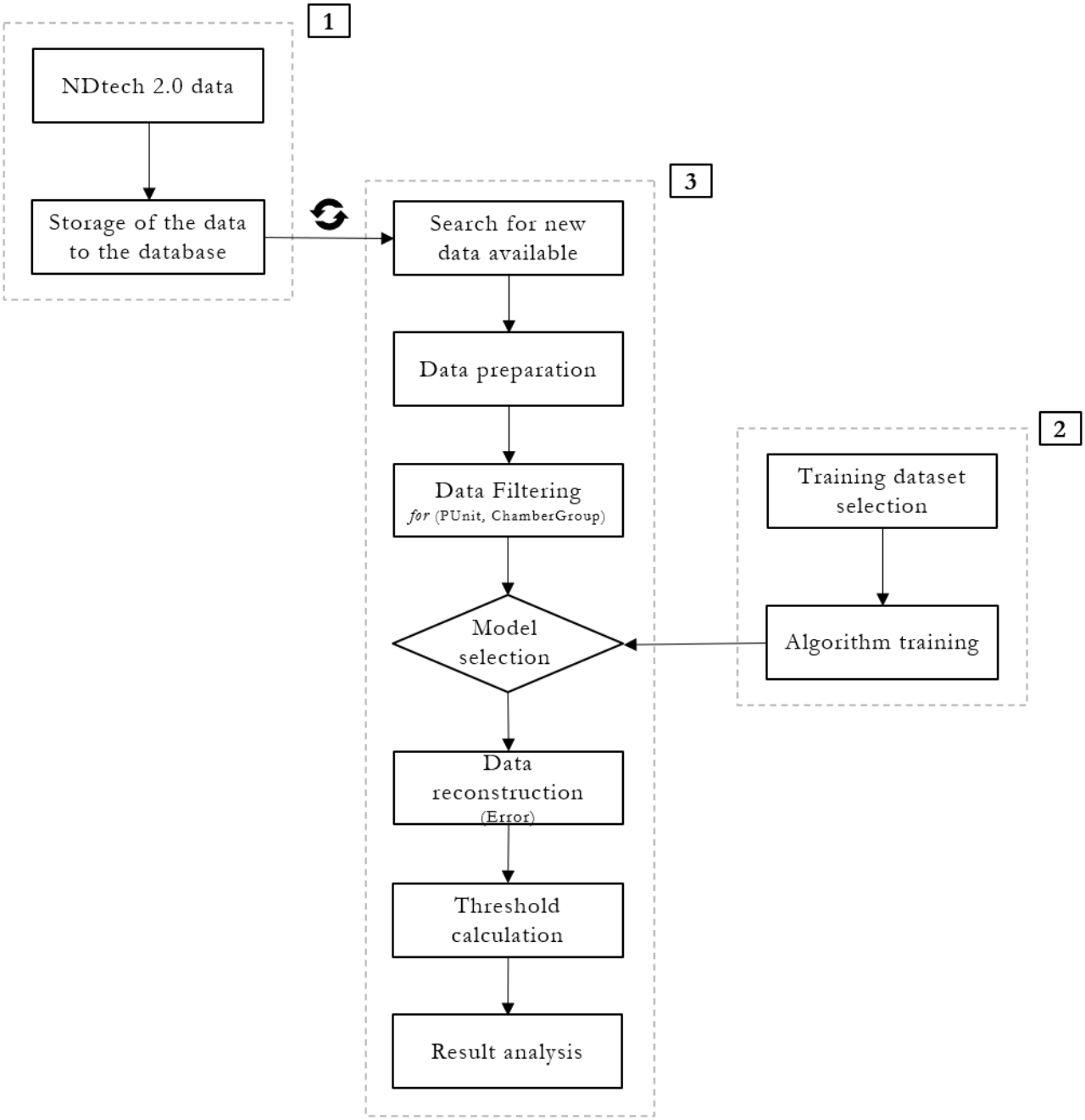


Figure 4.1 – Chart flow of the anomaly detection system

The data is recorded from the distinct processing units within the module are stored in the same database easing the process to retrieve it. Therefore, the data can be obtained and scaled covering the instances for the entire module, and later be filtered to the certain subgroup being analysed. Besides the data scaling, the data preparation phase also includes dealing with the potential presence of

missing values as have been specified during the exploratory data analysis. The training of the algorithms is individually set up in advance, where the train dataset is selected for each subgroup and later the saved models can be called when required during the analysis process.

4.2 NDTECH 2.0 MALFUNCTIONS

The reflection of the failures on the data and the depth of the knowledge that can be extracted from it, dictate the potential success of any anomaly detection algorithm. Until the moment of this work, the data registered from NDTech technology was mostly an unexplored and unknow field, specifically the impact of some of the known system anomalies on the data information and distribution. One of the intentions of this study was to raise a better understanding of how the most common reasons for stoppage, or system malfunctions detected by NDTech's current control would reflect on the data retrieve from the machine.

The current control procedures in NDTech detect the presence of an abnormal behaviour or failure based on the quality of the results of TCA detected from a contaminated sample while the AD algorithm will directly assess the system function from the descriptive information from all the variables available. Therefore, the effectiveness of anomaly detection algorithms heavily relies on the presence of distinct variations or patterns within the data during failures or abnormal behaviours. If a malfunction does not lead to noticeable changes in the data, the anomaly detection algorithm can struggle to identify and detect the underlying issue.

4.2.1 SIMULATIONS

By study the reflection of failures on the data facilitates a deeper understanding of the underlying dynamics of the technology and what needs to be looked at. To reach it, malfunctions simulations were implemented by intentionally inducing failures in a controlled environment, during a predetermined interval, to check how the technology data behaves with the presence of abnormal situations. The goal was to replicate some of the most common reasons for stoppage in NDTech 2.0, due to their interference with the TCA detection quality, to verify if these distortions would also be visible and detected on the data.

A set of different known failures that could be replicated, from problems in the incubation chambers, restrict or disable the gas circulation into specific processes or constrain the normal function of the columns were some examples of the total of 25 simulated problems. Depending on the components or sections affected by the failure, the effect could be extended to more than one of the identified subgroups, as there are many shared processes. If the problem is located at a specific chamber or one of the analytical columns, it is an individual problem of the chambers group and PUnit where it belongs, while problems with any of the valves, involve and impact the both the even and odd chamber group of that PUnit.

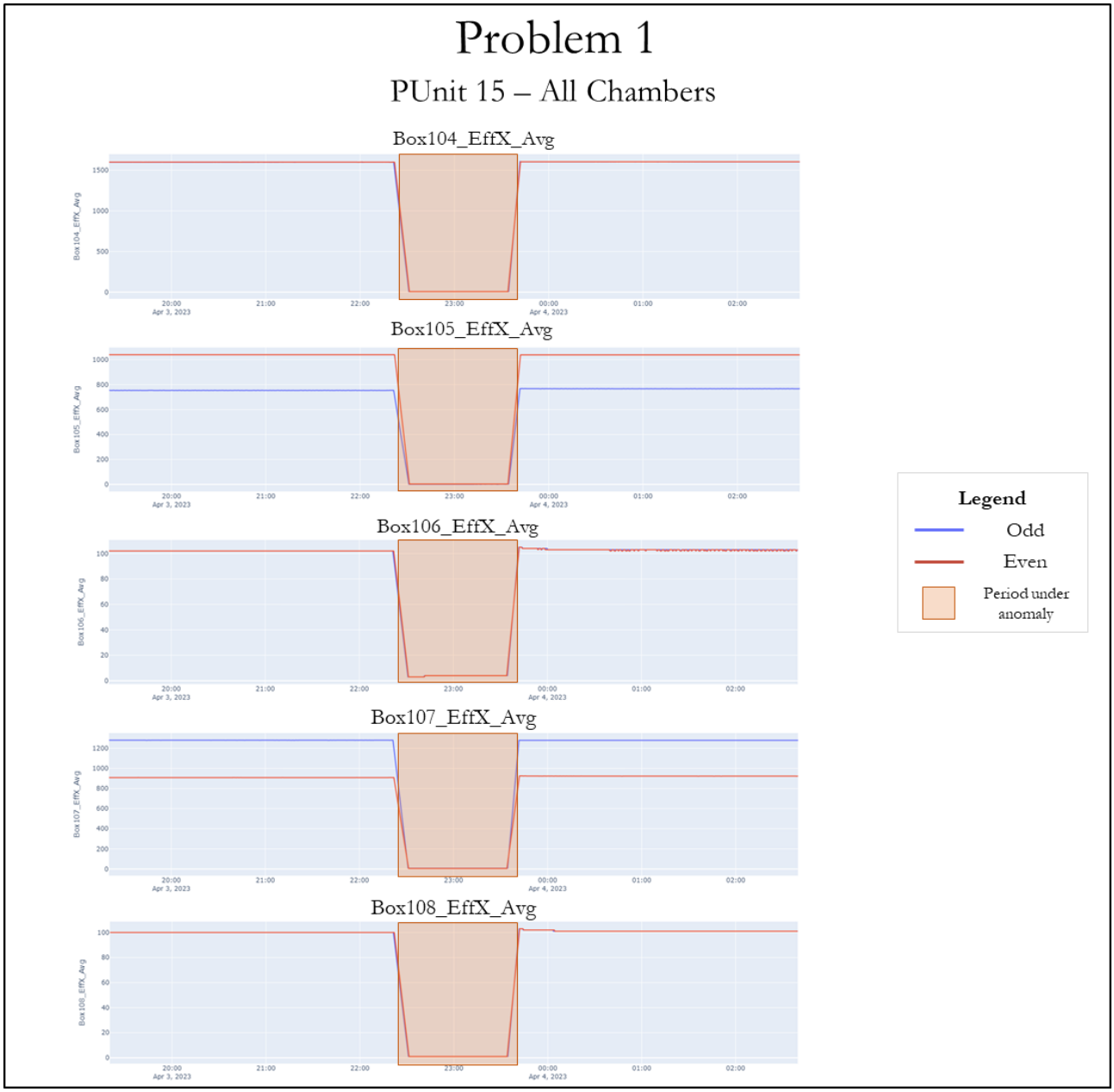


Figure 4.2 – Plots with the average's values of the *Box* (B104, *B105*, *B106*, *B107*, *B108*) variables from both chamber groups of PUnit 15 during the surrounded period of an induced malfunction on the nitrogen injection to the system

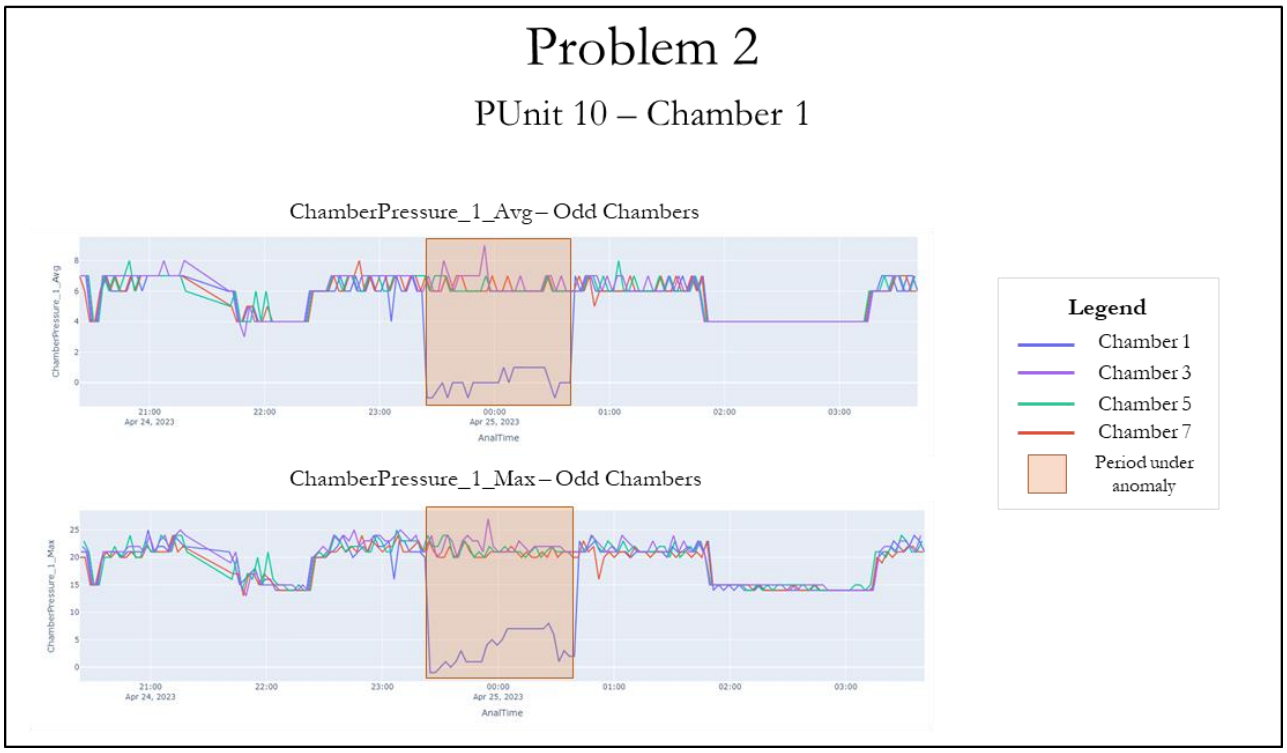


Figure 4.3 - Plots with the average and maximum values of the *Chamber_Pressure* variable from the odd chambers from PUnit 10 during the surrounded period of an induced malfunction in the chamber 1 of the PUnit 10

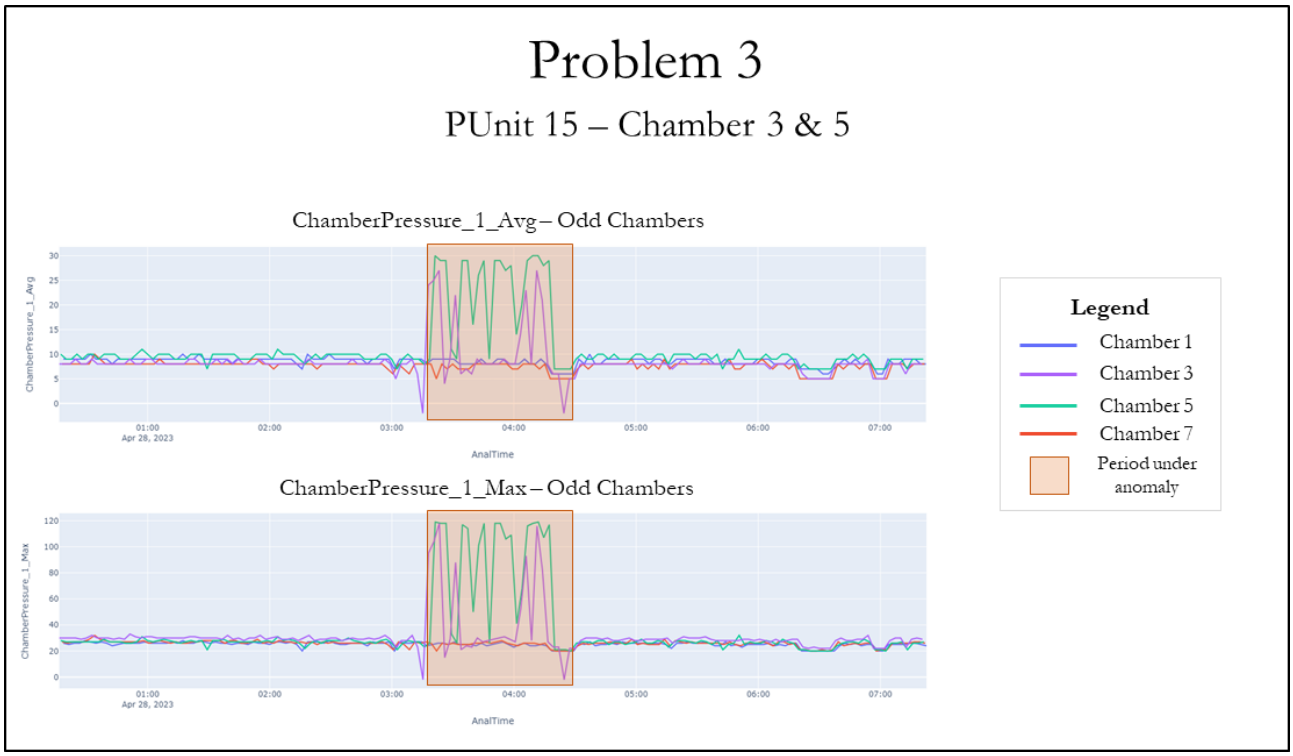


Figure 4.4 - Plots with the average and maximum values of the *Chamber_Pressure* variable from the odd chambers from PUnit 15 during the surrounded period of an induced malfunction in the chamber 3 and 5 of the PUnit 15

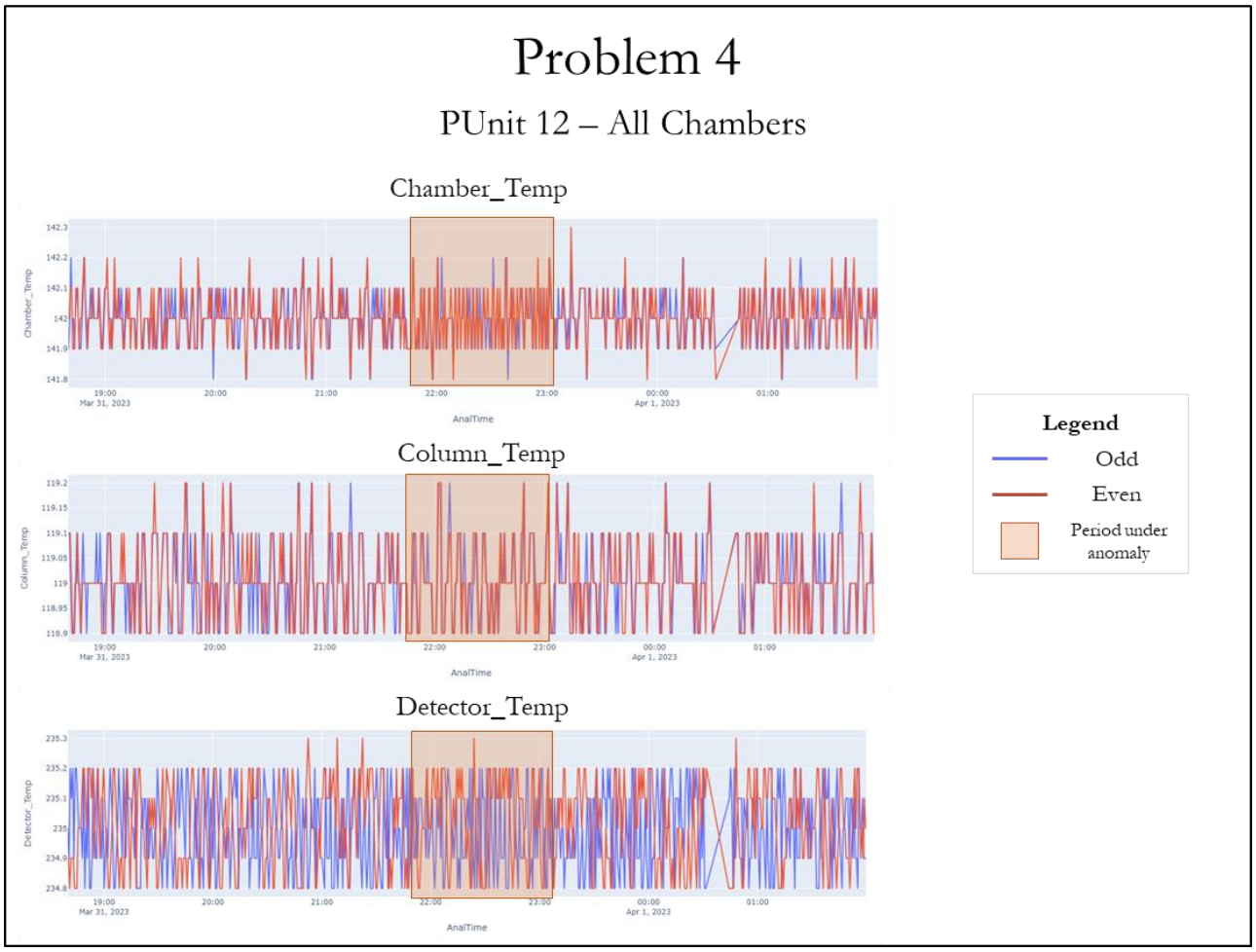


Figure 4.5 - Plots with temperature registered from the incubator chambers, columns and detector for both chamber groups of PUnit 12 during the surrounded period of an induced malfunction related to temperature changes in the PUnit 12

After analysing the different failures and the underlying data, it was possible to find distinct patterns over the impact of each simulation (*Figure 4.2* - *Figure 4.5*). Among the observations, certain situations displayed consistent shifts in expected features throughout the entire simulation, just like in problem 1 and 2 (*Figure 4.2* and *Figure 4.3*). In contrast, other situations showed fluctuations in the values of specific variables, which problem 3 (*Figure 4.4*) is an example. The features affected were linked to the nature and location of each malfunction. In instances where issues concerned to chamber-related components, the chamber pressure were the only conditioned feature (problem 2 and 3). Conversely, anomalies tied to certain valves functions exhibited a broader impact, disturbing the behaviour of multiple variables, mainly the *Box* features (problem 1). Finally, some of the induced

failures demonstrate having any type of impact on the data registered under the simulation period, as no variations or disturbance on the data could be spotted (problem 4, *Figure 4.5*).

Although the induced anomalies did not encompass full spectrum of potential issues faced by NDTech, they provide the insights required to understand the different nature of the anomalies and their reflection in the data. Which observations were affected, the extent of data variance induced by each malfunction, and which variables were affected, are the main factors that can dictate the difficulty to detect each anomaly. The data affected by anomalies were often contingent on the specific malfunction, with certain malfunctions producing effects on specific observations (based on the incubation chamber), while others had a broader influence. The degree of data variation resulting from failures can vary significantly. Some anomalies induced constant deviations in data, while others caused bumpy fluctuations. As for the number of variables influencing differentiation among malfunctions, it was found that this varied case by case. Some malfunctions affected a single variable, while more complex anomalies influenced multiple features.

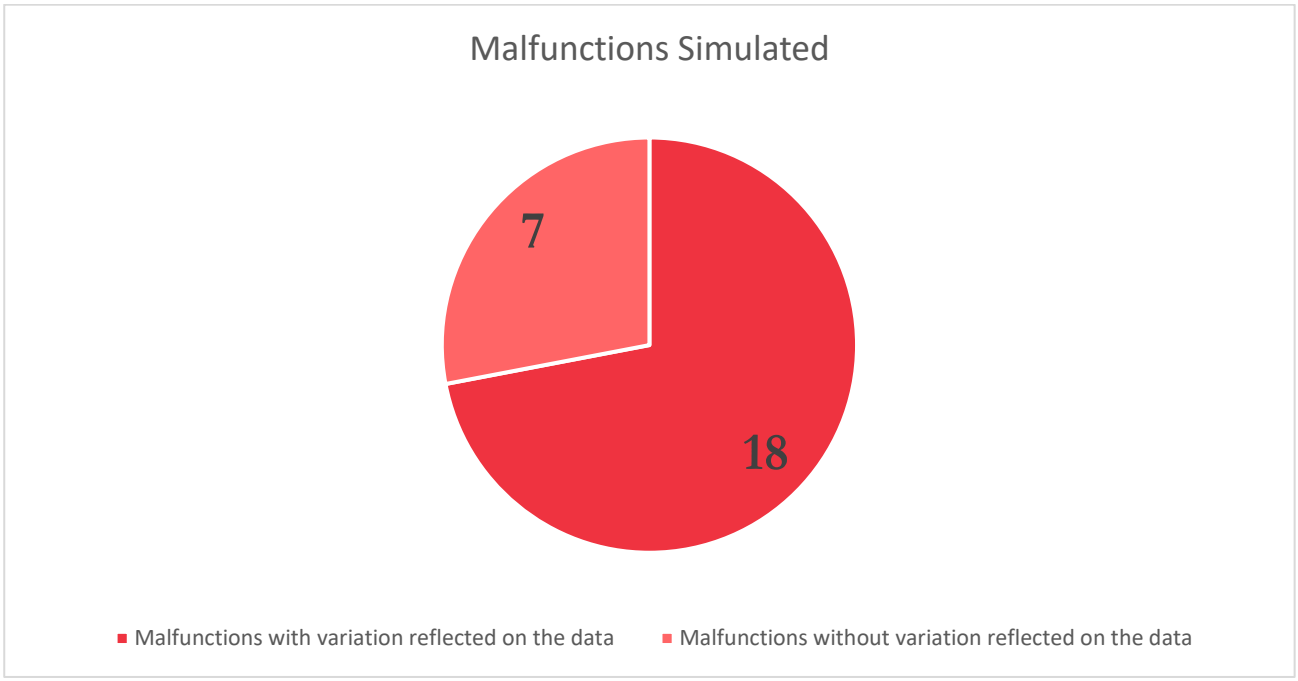


Figure 4.6 – Pie chart with the count of malfunction with and without reflections on the data.

It's worth noting that not all caused failures produced apparent impacts on the registered data during the simulation period. In these cases, the absence of variations or disturbances in the data prevent the

identification of anomalies. From the 25 simulations conducted, almost 30% of the anomalies displayed no detectable variations in the data (*Figure 4.6*). This finding highlights an important subset of instances where the anomaly detection algorithm could face serious challenges in recognizing anomalies due to the lack of conclusive evidence within the data.

4.2.2 ANOMALIES DETECTED

Under the context of the NDTech data system, where with the absence of labelled data, traditional metrics like accuracy and precision are unavailable. Malfunction simulations can, therefore, emerge as a vital and unique tool to have an idea of an algorithm effectiveness. The induced anomalies under a controlled environment, enable the exploration of the algorithm reactions to the variations that these malfunctions cause within the data. These observations registered under the simulated conditions act as anomalies replacements without requiring true labels to the data, they are the closest to a labelled dataset as its possible.

The utilization simulations extend beyond providing insights regarding the system and its corresponding reflection on the data. In this particular context, simulations serve a twofold purpose. Firstly, the induced failures offer a unique opportunity to examinate the algorithm's reactions and its sensitivity to variations in the data. This insight facilitates the refinement of algorithmic components, enabling them to accurately capture even subtle deviations from normal behaviour. Secondly, serve as a helpful benchmarking tool. They provide a reference point against which the algorithm's capacity to differentiate between normal and abnormal observations can be measured. Furthermore, simulations permit the adjustment of parameters such as thresholds to improve the performance.

In the following figures (*Figure 4.7* - *Figure 4.10*) are some examples of the results obtained from the analysis of the induced problems, representative of the overall performance of the models, with the inclusion of both positive situations where the models delivered the desired outcome as well scenarios where the models lack the capability to detect the anomalous observations.

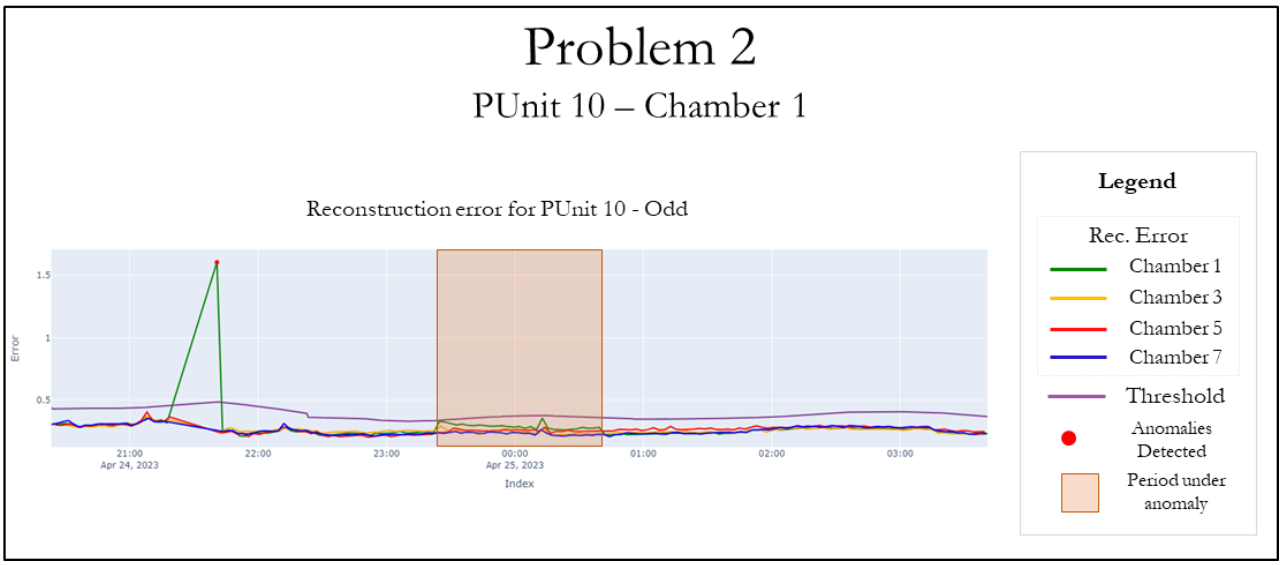


Figure 4.7 – Plot with the model's reconstruction error for the odd chambers from PUnit 10 during the surrounded period of an induced malfunction in the chamber 1 of the PUnit 10

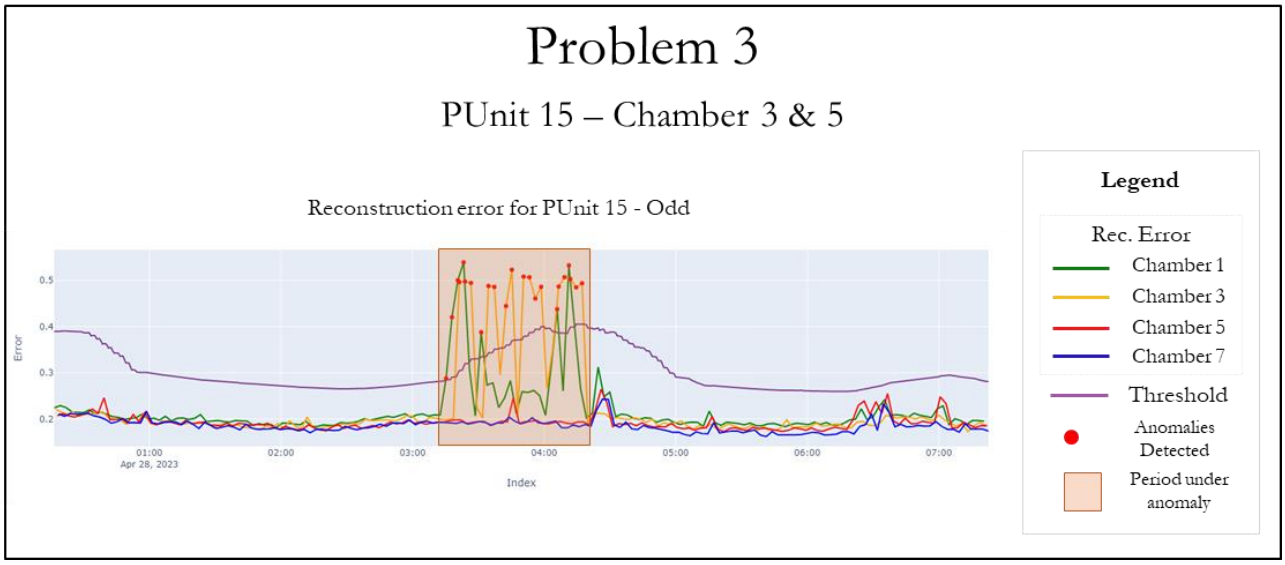


Figure 4.8 - Plot with the model's reconstruction error for the odd chambers from PUnit 15 during the surrounded period of an induced malfunction in the chamber 3 and 5 of the PUnit 15

In scenarios where the failure simulated caused a decreased in the chamber pressure (illustrated on problem 2, *Figure 4.7*), the LSTM autoencoders failed to detect any anomaly as the reconstruction error suffered a very minor increase on the instances analysed on the affected chamber over the rest, indicating that the model could recreate the anomalous observations almost as precisely as the rest.

This situation, probably derived from the training process, resulting in the opposite behaviour from the expected, and after some analysis it was found that there were more time periods where the data, more precisely, the chamber pressure feature, had similar values from the observed during the simulations, although they were not enough to influence the TCA analysis to the extent of being detected during the manual controls. Alternatively, the reaction of the models during malfunctions causing an increased pressure on the incubator chambers where the reconstruction error suffered a significant increase to the point of exceeding the established limits and being identified as anomalies, as illustrated in problem 3, *Figure 4.8*.

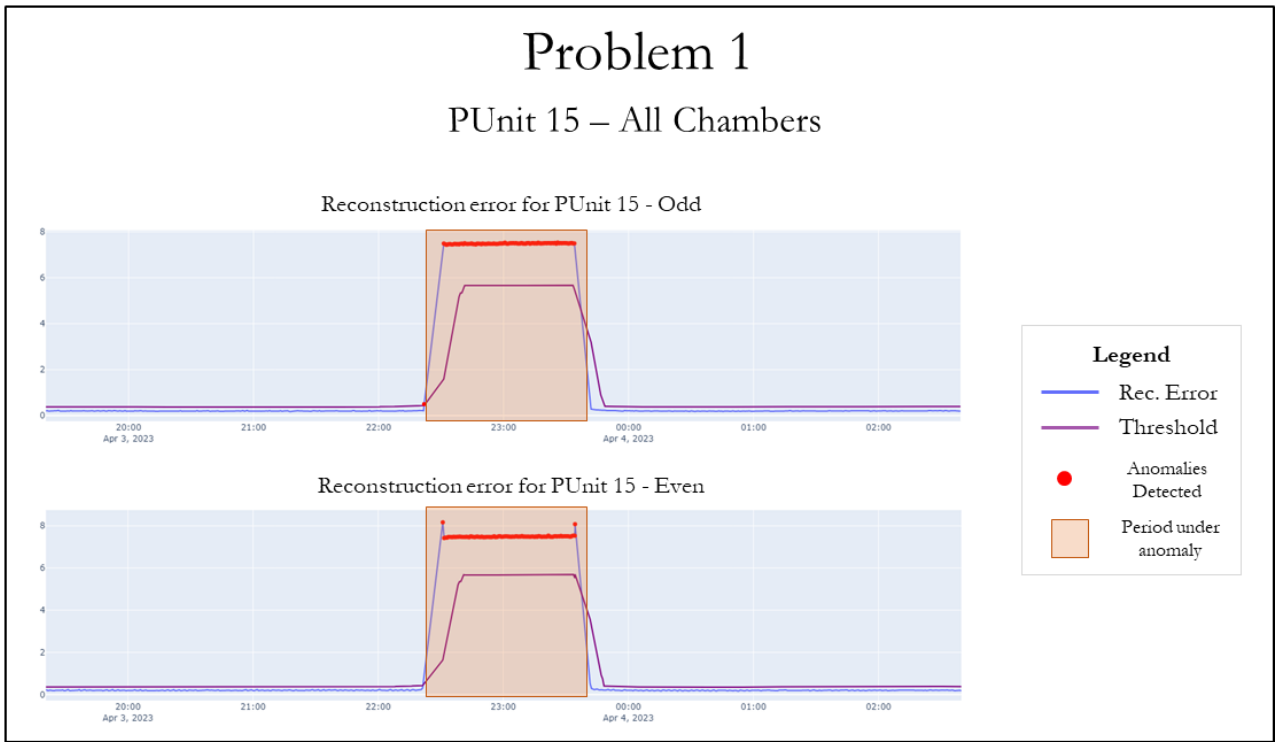


Figure 4.9 - Plot with the model's reconstruction error for chambers groups from PUnit 15 during the surrounded period of an induced malfunction on the nitrogen injection to the system

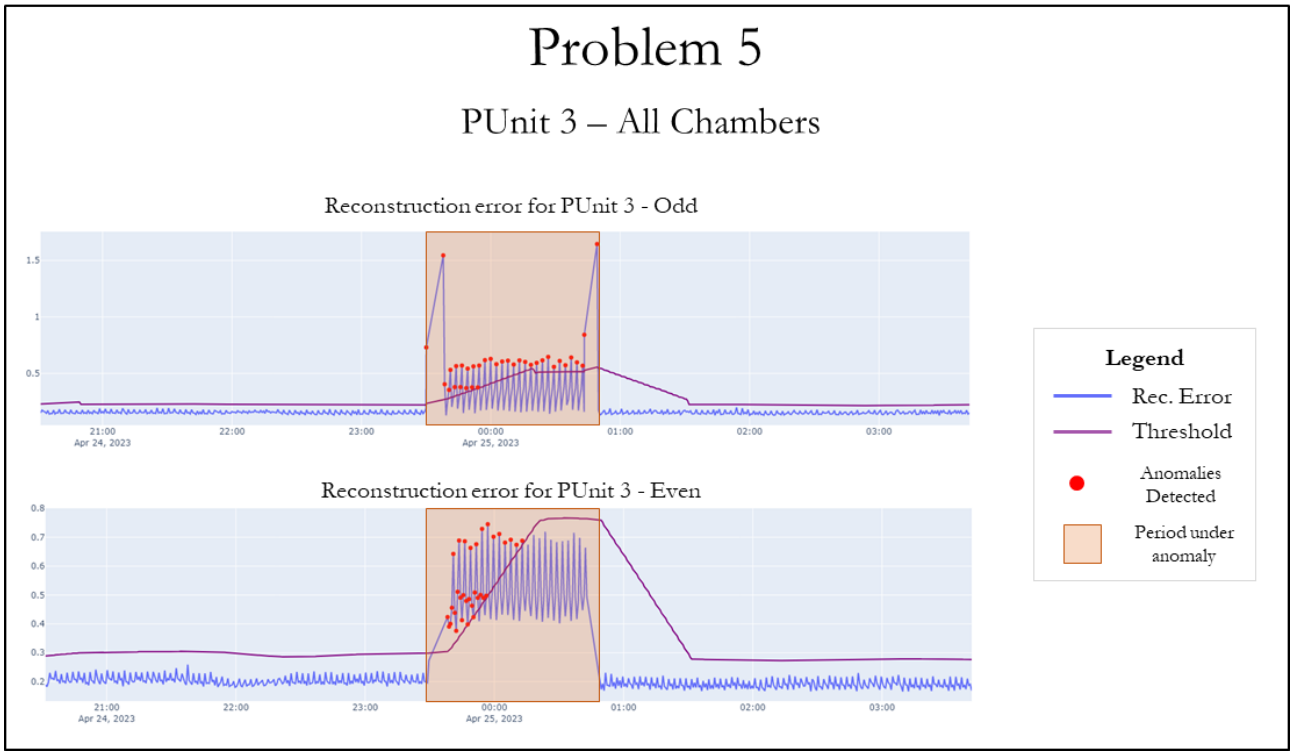


Figure 4.10 - Plot with the model's reconstruction error for chambers groups from PUnit 3 during the surrounded period of an induced malfunction on the initial position from the Stream Selector valve.

Regarding the cases where the affected features were linked to the valve's operation, mainly the pneumatic transport systems responsible for controlling nitrogen injection (*Box* features), the autoencoders exhibited a greater difficulty to accurately recreate the abnormal instances resulting in a generalized escalation in reconstruction errors. The degree of this discrepancy correlated directly with the number of features impacted, which becomes evident from the results of problem 3 and 5 (*Figure 4.8* and *Figure 4.10*). Specifically, in problem 3, a greater number of features were influenced during the simulation compared to problem 5, leading to significantly higher errors from the models. The extent of the discrepancy between the original input and the model's recreated output, reflected the time required for the threshold to adjust to the simulated situation. In instances where the range of fluctuations was narrower, this adjustment occurred more swiftly, leading to the prompt acceptance of the newly acquired values as normal, which the is illustrated an example on problem 5 for the subgroup Punit 15 – Even chambers.

Table 4-1 – Number of observations under conditioned operations

NUMBER OF INSTANCES UNDER MALFUNCTIONS	NUMBER OF DETECTED ANOMALIES
1593	827
	51.91%

Upon reviewing the total number of the observations recorded during faulty operations, as presented in Table 4-1, its perceivable that the models identified over 50% of the anomalous instances. However, this value by itself does not represent the true potential of the developed anomaly system. It is crucial to consider the specific circumstances introduced earlier that directly influenced the percentage of detected anomalies. In certain situations, the adaptive threshold adapted rapidly to anomalies, restraining the recognition off all instances as anomalous. Additionally, in some cases the effects of malfunctions were not continuous over time. It is conceivable that during certain periods within the simulation, some observations reflected normal system operations. Finally, the recognized inability from the models to distinguish the decreased pressure on the system had a significant impact on the final figures. If we exclude instances under lower pressure scenarios, the percentage of detected anomalies would reach approximately 70%.

The overall results from the induced malfunctions were positive, acknowledging the model's ability in more accurately reproducing the normal data. However, there were instances when this alone was insufficient to raise suspicions about abnormal behaviour within the system. This was most evident for the simulations causing a decrease in the chamber pressure, that were a significant portion of the tested malfunctions. It's important to recognize that while simulations provide a controlled environment, the real world can often be more complex and unpredictable. Therefore, the insights gained from simulations should be interpreted as a foundation for algorithmic improvement rather than a guarantee of real-world performance.

4.3 ONLINE ENVIRONMENT

The transition into a dynamic online system driven by the necessity to monitor in real-time data streams demand adjustments in various aspects from traditional static problem methodologies. In

contrast to a static system, where the data is collected in advance and remain throughout the process, in a continuous environment new data must be collected from the database. This change requires an efficient mechanism capable of ensuring the most recent data to be fed to the system as soon as possible. The same applies for the remaining processes that need to adapt to this new reality, the data preparation, data scaling and threshold calculation must adjust to the new face challenges and particularities involving a real-time analysis.

This section will outline the modifications made to the previously system architecture introduced to enable the implementation of an autonomous anomaly detection system capable of analysing the new data arriving from NDTech 2.0 as well as the results obtained from it. Additionally, a graphical user interface concept will be introduced. This interface could potentially be utilized by NDTech collaborators onsite, enabling them to visually observe the information provided by the anomaly detection system in real time, enabling its practical utilization.

4.3.1 SYSTEM ARCHITECTURE

The NDTech 2.0 data is store in a SQL Server with a refresh rate of 5 minutes, where the blocks of data analysed in these time intervals is registered every time the server is updated. Therefore, it was needed to create a system that when elapsed the defined time interval, checked the new data available and retrieve it to feed the anomaly detection algorithm. In the *Figure 4.11* below is a representative chart flow of the system architecture.

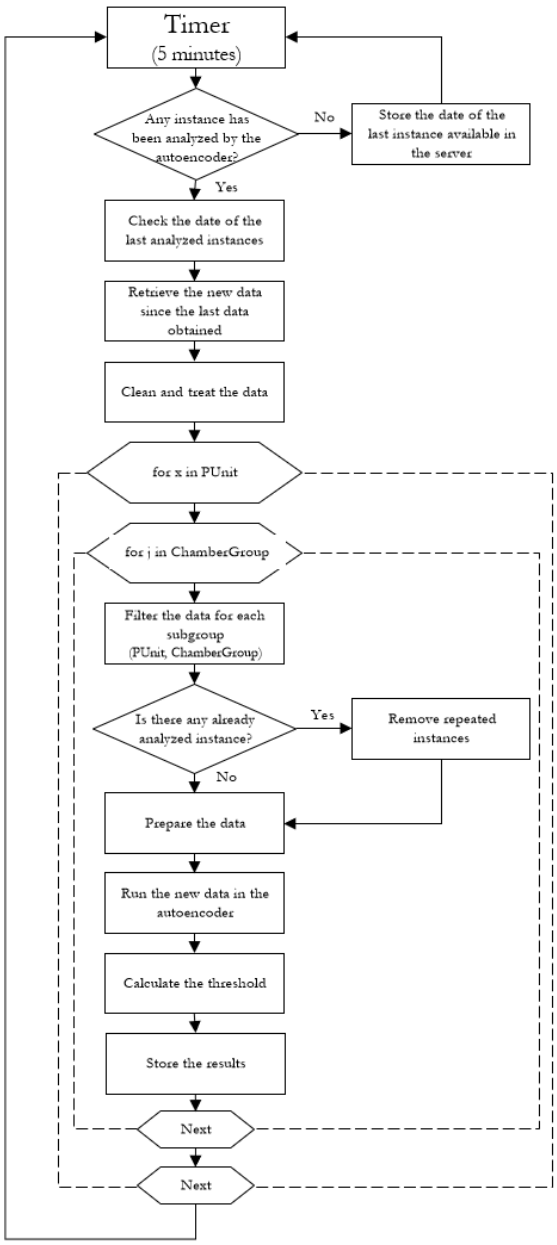


Figure 4.11 - Chart flow of the online anomaly detection system

To access and retrieve the data from the server, the Pyodbc Python module was utilized. This module establishes a connection with the database and search the new data available when required, based on the system's historical information. A timer feature was implemented, with an interval of 5 minutes, that defined the restart of the analysis cycle.

Two important tasks arriving with the addition of new data to the system ensuring that no instance is left without being passed through the correspondent autoencoder and to guarantee that no observation was scanned twice. To ensure that, the following processes were employed:

1. When searching for new available data the system checked for the oldest analysis date within all subgroups previous analysis. Afterward it retrieves all the available data from that point forward.
2. Later, after filtering the data to the certain subgroup, the system confirmed whether any of the selected instances have already been processed by the model previously. If such instances exist, they were discarded to avoid duplication.

4.3.2 RESULTS

The implementation and operation of the online anomaly detection system, which underwent testing for a duration of 15 days, present a unique challenge due to the absence of labelled classification, rendering it impossible to measure the model's performance. Nevertheless, this testing period offers valuable insights into the model's performance in a practical context. Firstly, it affirms the model's adaptability to near real-time analysis. Furthermore, it furnishes a comprehensive understanding of both the quantity and nature of anomalies identified by the system, offering crucial practical insights.

Table 4-2 – Table with the summary of the results from the online test

PUNIT - CHAMBERGROUP	TOTAL INSTANCES	ANOMALIES DETECTED	% ANOMALIES
PUnit 3 - Odd	29 412	178	0.61%
PUnit 3 - Even	29 396	163	0.55%
PUnit 10 - Odd	25 856	56	0.22%
PUnit 10 - Even	25 848	47	0.18%
PUnit 12 - Odd	30 444	206	0.68%
PUnit 12 - Even	30 424	110	0.36%
PUnit 15 - Odd	31 536	1 299	4.12%
PUnit 15 - Even	31 540	16	0.05%
	234 456	**2,075**	**0.89%**

The table above (*Table 4-2*) presents the statistics concerning the cork stoppers processed through the NDTech 2.0 module. In this dataset, the models identified 2,075 instances as abnormal out of a total of 234,456, which accounts for less than 1% of the entire dataset. These observations displayed deviations from the expected normal behaviour according to the underlying data, resulting in inaccurate reconstructions by the models. Notably, over half of these anomalies were concentrated within a specific subgroup, where the proportion of identified abnormal instances was significantly higher compared to the other subgroups.

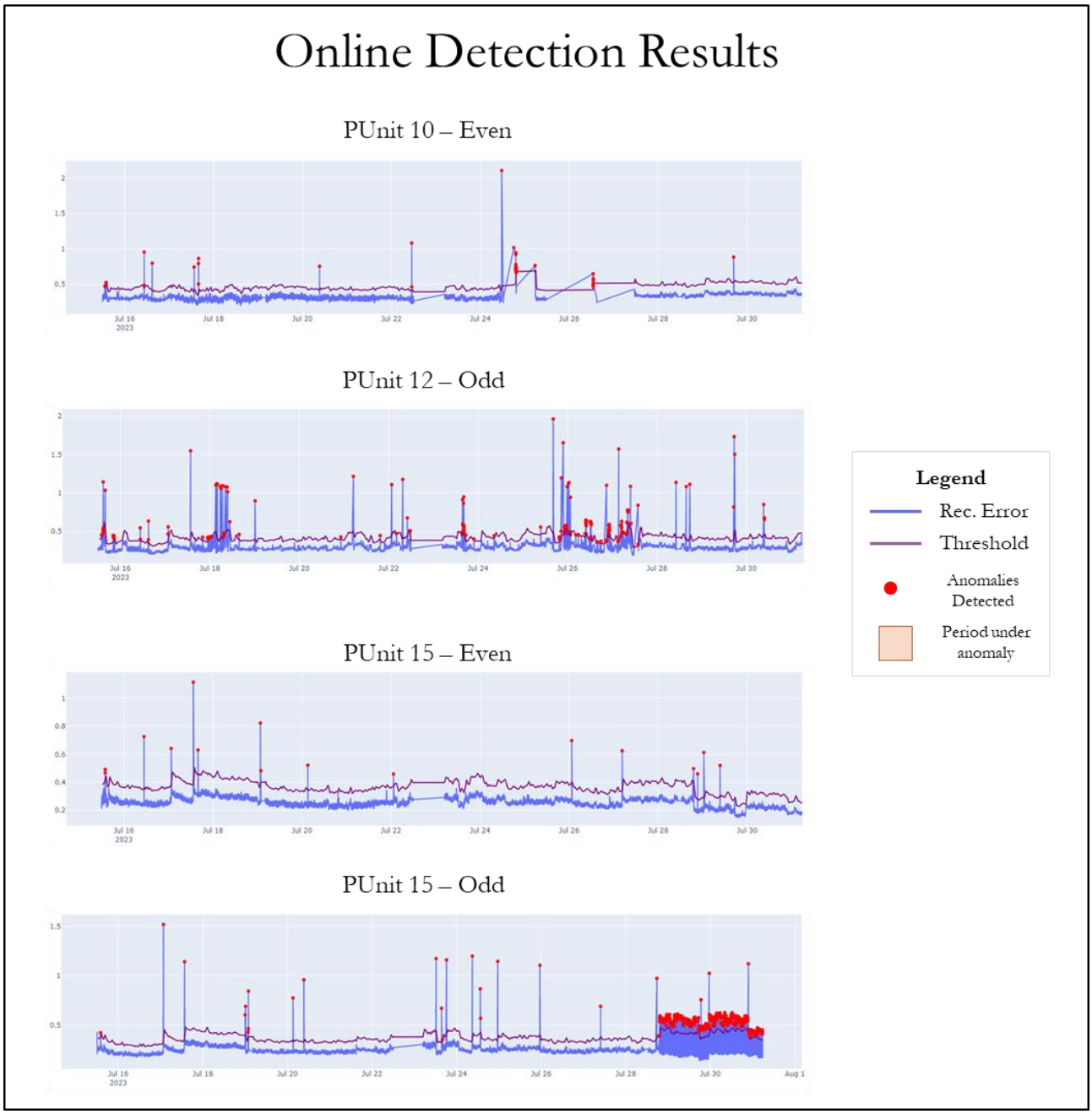


Figure 4.12 – Plots with the model's reconstruction error for some of the subsystems from the online tests

The recorded results from the system provide a valuable means to gain a deeper understanding of the identified anomalies. Certain recurring patterns occur across multiple subgroups evidencing some circumstances that triggered the anomaly detection, visible on the illustrated examples in *Figure 4.12*. A common scenario across all subsystems is the detection of isolated anomalies within the data, which can be explained by the high number of outliers present on the data perceived during the data

exploratory analysis. It's important to note that these isolated anomalies do not necessarily indicate abnormal functioning of the NDTech technology. Additionally, a notable finding that occurred on multiple occasions, is that when some of the processing units is temporarily interrupted for some reason, the subsequent shift in errors upon reinitialization can lead to the detection of anomalies.

Concerning scenarios where multiple anomalies are consistently detected, indicating a higher likelihood of an underlying technology malfunction, we will conduct a detailed examination of the time interval spanning from July 28, 2023, at 00:00 to July 31, 2023, at 05:40. This investigation will be particularly focused on the odd chambers in PUnit 15, where numerous abnormal observations were identified over most of this period. The *Figure 4.13*, provided below, illustrates that all the observations recorded from chamber number 3 were flagged as anomalous at a specific point in time due to a notable increase in the reconstruction error, while the remaining chambers remained stable.

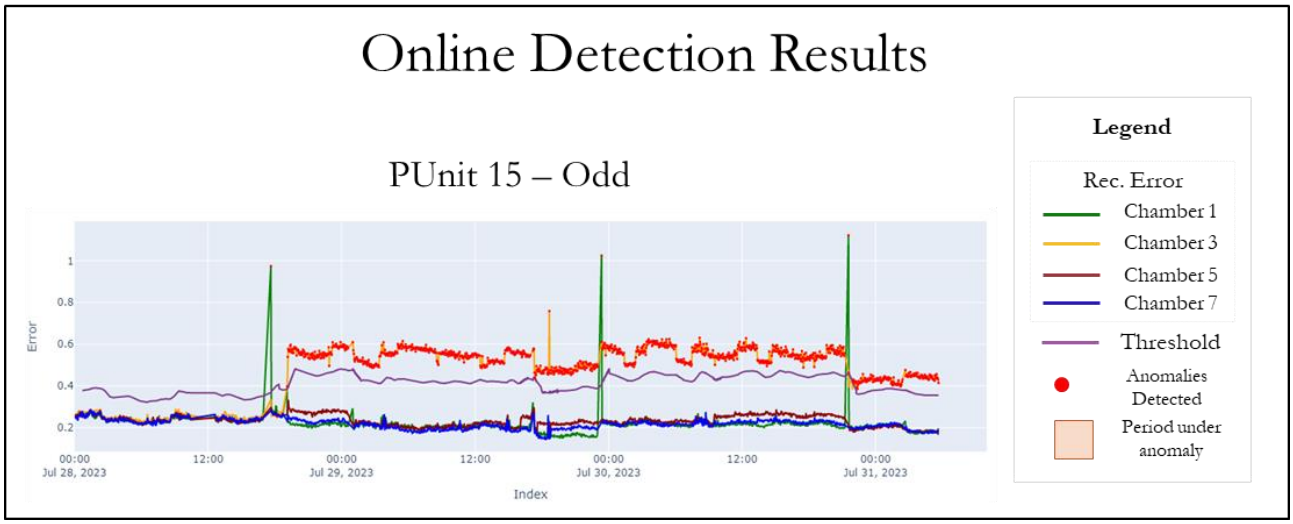


Figure 4.13 - Plots with the model's reconstruction error for the odd chambers from PUnit 15 for a selected interval from the online tests

Further investigations based on the discrepancies between each feature's projection from the model and the original input, facilitate the identification of the variables that contributed the most to the increased errors. During the selected time frame, the elevated error values were mainly caused to a significant error in the *ChamperPressure* recreation This observation aligns with the recorded chamber pressure values for the odd chambers during the specified time interval, as represented in the figure below (*Figure 4.14*), which confirm the increased pressure in the chamber number 3.

Notably, it's worth mention that these conditions remained consistent over an extended period and were not detected through multiple manual control procedures conducted by NDTech's collaborators. This suggests that there was no substantial impact on the TCA detection chromatography despite these anomalies.

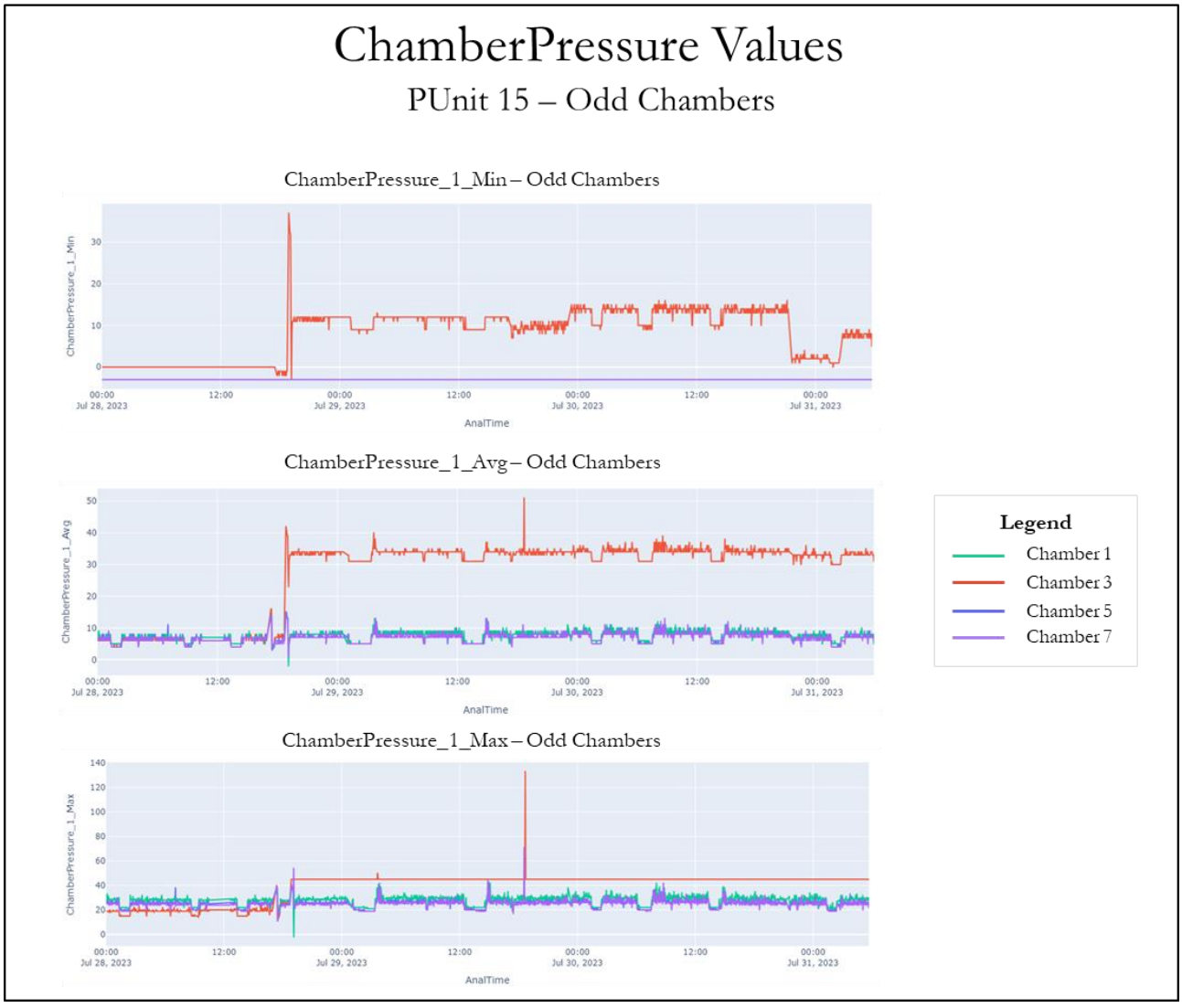


Figure 4.14 - Plots with the minimum, average and maximum values of the *Chamber_Pressure* variable from the odd chambers from PUnit 15 during the previously selected interval from the online tests

Based on the comprehensive analysis of the results obtained, both from the scrutiny of induced malfunctions and the online testing phase, it cannot be definitely asserted that both research questions, previously formed for this project, can be affirmatively answered. While the models were able to identify anomalies during the majority of the simulated problems, was not possible to establish

whether the anomalies detected during the online tests corresponded to true abnormal operations within NDTech. In essence, the models proved capable of recognizing certain deviations in the data caused by malfunctions under a controlled environment, with some exceptions. However, the inherent limitations of NDTech system prevented a conclusive extrapolation of these findings to real-world NDTech operations.

4.3.2.1 RESULTS VISUALIZATION

The primary objective of implementing an online detection system is to enhance the existing control procedures by enabling faster and more efficient detection of deviations from the technology's normal operation. To achieve this, besides the accurately recognition of true-positives anomalies, it is crucial to provide the pertinent information for the team onsite, so they can be aware of the current situation and take appropriate actions. To facilitate this, will be introduce a conceptual visualization interface. This interface is designed to present updated outputs generated from the anomaly detection system. With the help of the PySide 6 library were developed two distinct windows, each displaying different information. These visualization tools are intended to support the NDTech team to easily interpret the current operating scenario.

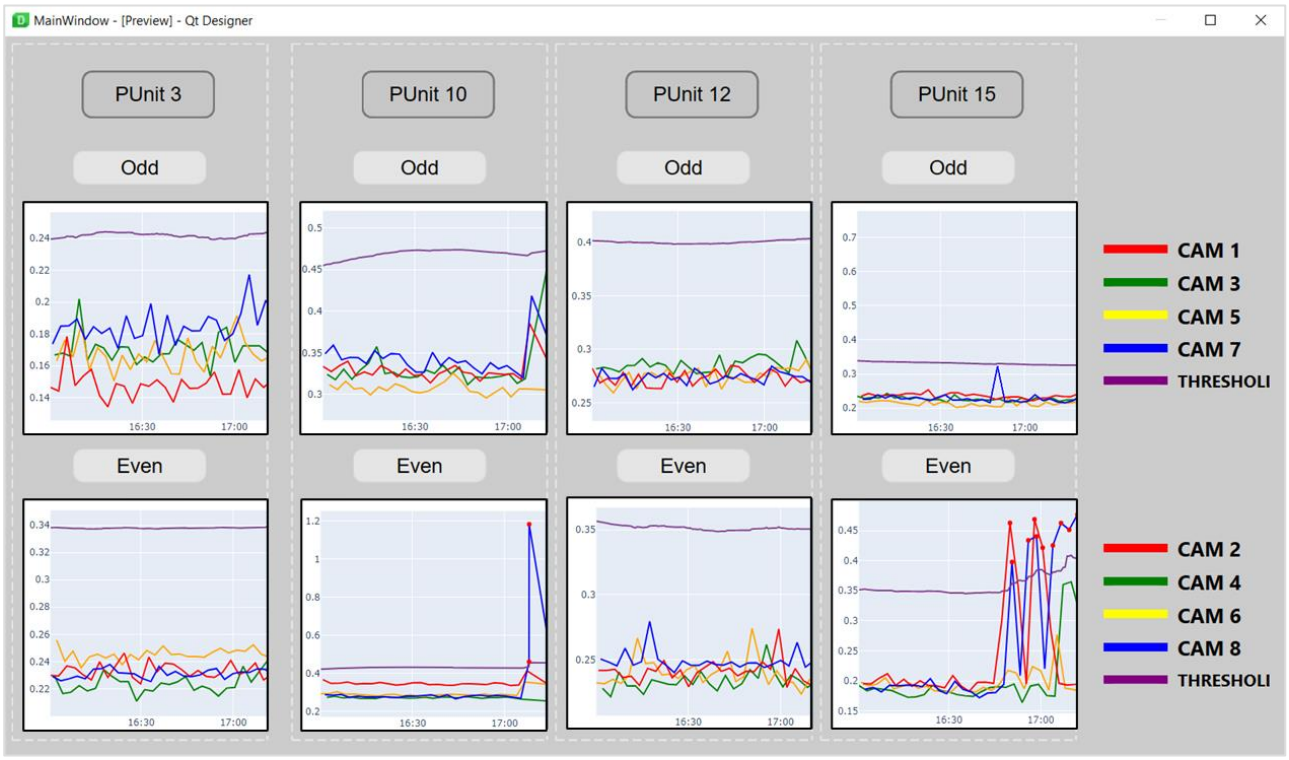


Figure 4.15 – Conceptual visualization window for the results from the online AD

The primary window, the "Main Window," will serve as the central hub for visualizing crucial data, featuring plots for each subgroup, displaying the reconstruction errors for each individual chamber, along with the estimated threshold values, as illustrated in *Figure 4.15*. This display facilitates the team to have an immediate understanding of the historical results from the system within a specified time frame. Notably, it enables the identification of observations that surpass the threshold line, indicating the detection of anomalies.

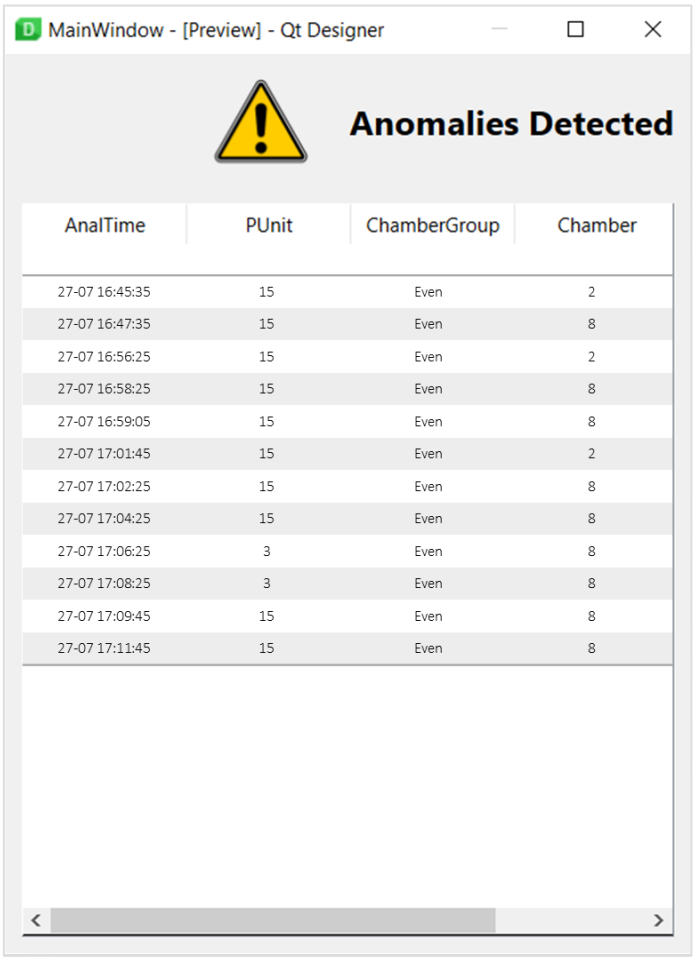


Figure 4.16 - Conceptual dashboard with a list of the anomaly detected from the online AD

Furthermore, a second window has been created, pictured in *Figure 4.16*. This window features a table listing the anomalies identified during the specified time interval from the previous window. It includes additional details such as the corresponding analysis time, the correspondent processing unit, as well as chamber and chamber group information for each anomaly. This provides a comprehensive overview of the detected anomalies, enhancing the team's ability to assess and address these issues effectively. To enhance the depth of information regarding the listed anomalies, it could be included the variables that had the most significant impact on the increased error. This addition would highlight the responsible variables that triggered the algorithm's alert, consequently facilitating the identification of potential sources of the problem.

It is worth highlighting the potential of the implementation of a visualization tool, as present in this report, featuring real-time information of the anomaly detection system implemented. This tool can

be crucial to help the on-site team in monitoring NDTech operations and confirming the accuracy of detected anomalies. The real-time insights enable a proactive approach and allows the team to quickly respond to potential issues. With the confirmation of a good performance from the anomaly detection system is possible to enhance the current overall effectiveness of the monitoring process and contribute to the continuous improvement of NDTech operations.

CHAPTER 5 FINAL CONSIDERATIONS

This summary encapsulates the process of project stages, highlights significant findings during its implementation, and concludes with the achievements of the proposed solution. Additionally, it mentions some considerations for future improvements that can enhance the system's effectiveness.

5.1 CONCLUSIONS

The sophisticated NDTech system demands close surveillance to ensure its optimal performance. Currently, this heavily relies on human-dependent procedures, resulting in time-consuming processes and limitations in effectiveness. The introduction of operational data accessibility on the second version of the technology opens the door to different approaches and potential opportunities to improve the current scenario. This project was undertaken to explore innovative methods for monitoring and identify deflections from the technology's normal operation. The proposed approach leveraged the power of machine learning techniques into an automated control system capable of overcoming the existing constraints.

The data exploration has confirmed the findings of previous research conducted by Rogério Nunes, which established variations in the operations among the distinct processing units. Moreover, it established the divergence from the incubator chamber groups within each PUnit. This scenario demanded an individual analysis of these subgroups due to the substantial separation between the information registered from their operation. Furthermore, the study on the available data provided essential information and insights for some of the preprocessing techniques applied before feeding the data into the anomaly detection algorithm. Addressing Null values and scaling the data, were two important stages that facilitate and enhance the algorithm performance, respectively.

One conditioning characteristic of this system’s environment is the absence of labels on the data, which had a substantial impact on both algorithm selection, and more importantly, the assessment of the model's performance. This absence of labels renders the task of evaluating the model's efficacy impossible. A noteworthy study conducted within this project involved intentionally induced

anomalies under a controlled environment to gain perceptions into how they reflect on the underlying data. With this experience it became evident that certain malfunctions did not exhibit detectable variations on the data. This highlighted the inability of the anomaly detection models in identifying such anomalies. The simulated problems served a dual purpose of revealing information regarding certain malfunctions but also established a benchmark for testing the performance and abilities of the detection models in identifying the periods under faulty operations. For anomaly detection task, was opted to use autoencoders, namely LSTM autoencoders. These models were chosen to combine the data reconstruction ability inherent in autoencoders and the sequential memory capabilities provided by LSTM layers. Each model to the identified subgroups within the dataset underwent a training and optimization process, to achieve the best possible alignment between the models and the corresponding dataset, using the raw data, where the potential presence of anomalies was a significant concern.

As previously mentioned, the simulations played a major role in assessing the model's ability to identify fluctuations in the data under the conditioned scenarios. In most cases, when anomalies were introduced, the reconstructed error increased exceeding the established threshold line. The nature and extent of the malfunction's impact on the data, namely the number of features affected and the range of deviations, directly influenced the identification of anomalous instances. Under some of the simulations where substantial errors were introduced due to a higher number of affected features, all instances within the defined interval exceeded the threshold. However, in scenarios where fewer features were affected, the adaptive threshold allowed the model to classify new reconstruction errors as normal. Moreover, it's crucial to mention that during malfunctions that caused a decrease in chamber pressure, the model's accuracy in recreating such instances did not significantly deteriorate. This results in only a slight increase in the discrepancy between the input and the reconstructed outputs, which wasn't substantial enough to trigger the anomaly alert.

Following the necessary adaptations within the preprocessing treatments of the data and the autonomous cyclic data retrieval, was conducted a comprehensive 15-day test of the online anomaly detection system. For the data scaling was implemented an incrementally approach to estimate the updated mean and standard deviation with the insertion of new data, while for the threshold calculation was used a sliding window method to estimate the required mean value of the current window. The developed architecture proved to be capable of analysing the new blocks of data within the 5-minute updating rate of the database. During the testing phase, while impossible to verify the

accuracy of detected anomalies, the percentage of anomalous instances fell within the expected range for the technology. Notably, in the online tests, one particular situation occurred to a certain odd chamber in processing unit 15. The model identified over 2 days of data as anomalous, in this chamber. However, manual controls performed during this period did not detect any changes in the TCA detection quality for the contaminated injected samples. To enhance the support for NDTech's on-site work monitoring, a concept for a visualization tool was introduced. This tool aims to provide real-time information on the anomaly detection system's results, offering a recent historic plot analysis and a detailed list with the detected anomalies identified to help locating the potential malfunction more effectively.

Overall, this project has significantly advanced the current understanding of NDTech's operational data and its potential to enhance control procedures effectiveness. The active involvement of the NDTech team in this project has promoted an appreciation for the importance of the recorded information, which had previously gone underutilized. While substantial progress has been made, there remains an opportunity to further expand the knowledge regarding each individual feature and its range of values. This knowledge could help clarify the scenarios encountered where the behaviours from specific features observed during the simulations were closely repeated during certain periods but went unnoticed by the current control procedures. Despite the challenges presented by unsupervised environments, the current project introduced an online detection system. When combined with a real-time visualization tool, the system has the potential to provide valuable information to support the workers on site. It represents a significant step towards optimizing control procedures and enhancing operational efficiency.

From Amorim's business perspective, this project leveraged the importance of transforming available data into valuable insights and decision-making information, an area that has generally been underdeveloped. With the extensive advanced technology infrastructure present within Amorim Cork's industrial facilities, there exists significant potential to greatly improve productivity and enhance overall process effectiveness by taking advantage of the available data. Specifically, the active involvement of the NDTech departments throughout the project has raised awareness regarding the utilization of NDTech's data towards control procedures. Finally, the project presented a potentially beneficial approach to support the current control system. However, the existing constraints of the available data prevented a full assessment of the model's abilities. Nevertheless,

with the implementation of the visualization tool along with additional online tests, it would be possible to establish whether the anomaly system could be beneficial or not.

5.2 FUTURE CONSIDERATIONS

The complexity and singularity of Amorim technology demands a customized solution for the proposed problem. In this context, the know-how from individuals working closely with NDTech on a daily basis can hold substantial potential to enhance the system's effectiveness. Given the current limitations, gaining a deeper understanding of how the anomalies reflect on the data, and developing a perception of expected behaviours under conditioned scenarios can greatly support data filtering for models training. The improved training can improve the model's ability to learn and accurately identify normal data during the training process. Moreover, exploring the connection between shifts in operation records and the quality of the TCA quantification process is crucial. During the development of this project were encountered situations where similar features behaviours from the malfunction's simulations occur during long periods of time without being detected by the current control process. Therefore, it's important to determine whether these shifts effectively impact on the system's operation and whether they should be detected by the anomaly detection system, or if they are acceptable values for the system's proper functioning.

Throughout the project's development, obtaining a labelled dataset was inaccessible, which presented unique challenges. However, it's worth noting that having a labelled dataset could potentially lead to a different stage of this problem. Such dataset would provide the base for conducting accurate performance assessments, offering exact measurements of the model's capability to detect the desired anomalies. This information could be beneficial for testing various methods and techniques, enabling an investigation of the best suited approaches for this particular problem. It would open new paths to both assessing the true potential of such control procedures and, refining anomaly detection strategies and optimizing model performance, leading to advances to the final goal of improving the current scenario.

CHAPTER 6 REFERENCES

Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. Neurocomputing, 262, 134-147.

Ahmed M., Mahmood A., Hu J. (2016). A survey of network anomaly detection techniques, Journal of Network and Computer Applications, Volume 60, Pages 19-31, ISSN 1084-8045. https://doi.org/10.1016/j.jnca.2015.11.016.

Baldi, P. (2012, June). Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML workshop on unsupervised and transfer learning (pp. 37-49). JMLR Workshop and Conference Proceedings.

Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. arXiv preprint arXiv:2003.05991.

Bhuyan M. H., Bhattacharyya D. K. and Kalita J. K. (2014). "Network Anomaly Detection: Methods, Systems and Tools," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303-336.

Edgeworth, F. Y. (1887). On discordant observations. Philosoph. Mag. 23, 5, 364–375.

Goldstein M, Uchida S (2016) A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. PLOS ONE 11(4): e0152173. https://doi.org/10.1371/journal.pone.0152173

Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: state of the art, challenges, and opportunities. ACM SIGKDD Explorations Newsletter, 21(2), 6-22.

Iglewicz, B., and Hoaglin, D. How to detect and handle outliers, american society for quality control. Milwaukee, WI, 1993.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, 448-456.

Ji, Wenying & Abourizk, Simaan. (2018). Simulation-Based Analytics for Quality Control Decision Support: Pipe Welding Case Study. Journal of Computing in Civil Engineering. 32. 05018002. 10.1061/(ASCE)CP.1943-5487.0000755.

Leithead, A. (2022). Why cork is making a comeback, BBC, https://www.bbc.com/future/article/20221013-why-the-humble-wine-cork-is-being-reinvented, accessed on 15th of January, 2023.

Li, P., Pei, Y., & Li, J. (2023). A comprehensive survey on design and application of autoencoder in deep learning. Applied Soft Computing, 110176.

Mobtahej, P., Zhang, X., Hamidi, M., & Zhang, J. (2022). An LSTM-Autoencoder Architecture for Anomaly Detection Applied on Compressors Audio Data. Computational and Mathematical Methods, 2022.

Nasser, A., Hamad, D., & Nasr, C. (2006). Kernel PCA as a visualization tools for clusters identifications. In Artificial Neural Networks–ICANN 2006: 16th International Conference, Athens, Greece, September 10-14, 2006. Proceedings, Part II 16 (pp. 321-329). Springer Berlin Heidelberg.

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. Ieee Access, 9, 78658-78700.

Nguyen, H. D., Tran, K. P., Thomassey, S., & Hamad, M. (2021). Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. International Journal of Information Management, 57, 102282.

Nunes, R. (2022). Eficiência operacional na deteçãao individual de TCA em rolhas naturais e de champanhe: uma análise estatística. FCUP.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. ACM computing surveys (CSUR), 54(2), 1-38.´

Pereira, J., & Silveira, M. (2018, December). Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 1275-1282). IEEE.

Principi, E., Rossetti, D., Squartini, S., & Piazza, F. (2019). Unsupervised electric motor fault detection by using deep autoencoders. IEEE/CAA Journal of Automatica Sinica, 6(2), 441-451.

Provotar, O. I., Linder, Y. M., & Veres, M. M. (2019, December). Unsupervised anomaly detection in time series using lstm-based autoencoders. In 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT) (pp. 513-517). IEEE.

Rettig, L., Khayati, M., Cudré-Mauroux, P., & Piórkowski, M. (2019). Online anomaly detection over big data streams. Applied Data Science: Lessons Learned for the Data-Driven Business, 289-312.

Rodríguez, Maria Luisa & Recio, Eliseo & Coque, Juan-José. (2009). The analysis of natural cork stoppers in transversal sections as an effective tool to determine the origin of the taint by 2,4,6-trichloroanisole. European Food Research and Technology.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science.

Russell, S. J. 1., & Norvig, P. (1995). Artificial intelligence: a modern approach. Englewood Cliffs, N.J., Prentice Hall.

Shölkopf B., Smola A.J., "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond", the MIT Press, Cambridge, Massachusetts, London, England, (2002).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

Tarasov, A., Cabral, M., Loisel, C., Lopes, P., Schuessler, C., & Jung, R. (2022). State-of-the-Art Knowledge about 2,4,6-Trichloroanisole (TCA) and Strategies to Avoid Cork Taintin Wine. In A. Morata, I. Loira, & C. González (Eds.), Grapes and Wine. IntechOpen.

Thakran, T. & Toshniwal D. (2012). Unsupervised outlier detection in streaming data using weighted clustering. *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, Kochi, India, pp. 947-952, doi: 10.1109/ISDA.2012.6416666.

Torralba, A., & Efros, A. A. (2011, June). Unbiased look at dataset bias. In CVPR 2011 (pp. 1521-1528). IEEE.

Tukey, J. W. (1977). Exploratory data analysis (Vol. 2, pp. 131-160).

Varun C., Arindam B., and Vipin K. (2009) . Anomaly detection: A survey. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

Wang, H., Peng, M. J., Miao, Z., Liu, Y. K., Ayodeji, A., & Hao, C. (2021). Remaining useful life prediction techniques for electric valves based on convolution auto encoder and long short term memory. ISA transactions, 108, 333-342.

Welford, B.P. (1962) Note on a Method for Calculating Corrected Sums of Squares and Products. Technometrics, 4, 419-420.