University of South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2023
Industrial IT and Automation

Data Analysis and Modelling of Induction Hardening Processes

Stefano Fernando Panzeri Reyes

Faculty of Technology, Natural Sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course:	FMH606 Master's Thesis 2023
Title:	*Data Analysis and Modelling of Induction Hardening Processes*
Pages:	*108*
Keywords:	*data science, exploratory data analysis, correlations, machine learning, neural networks, LSTM, induction heating, induction hardening, predictive maintenance*
Student:	*Stefano Fernando Panzeri Reyes*
USN Supervisors:	*Håkon Viumdal (Main) and Ru Yan (Assistant)*
Company Supervisors:	*Bjørn Sverre Aspheim (Progress) and Audun Hystad (Technical)*
External partner:	*ENRX AS*

Summary:

In collaboration with ENRX, this project conducted a technical feasibility study focused on data collection and analysis. The project involved gathering operational data from induction hardening machines owned by various company clients. This was collected by designing and installing a data-logging system on machines belonging to three of the clients. The collected data went through an initial phase of exploratory data analysis before being utilized to construct an LSTM (Long Short-Term Memory) Neural Network for error prediction.

During the first part of the project, the collected data sets were examined to reveal both linear and non-linear correlations. This analytical study provided valuable insights into how clients use the different machine parameters. It also identified variables that showed strong correlations, those that were less relevant to the specific case under investigation, and potentially missing variables that could be of interest.

The second phase of the project involved the creation of two LSTM models to predict a particular machine alarm. In the first case, the objective was to forecast the specified alarm by analysing the initial 10 seconds of a hardening process. The second case analysed the preceding five hardening processes to determine whether the next hardening process would encounter the targeted alarm. For both scenarios, the data required for input into the LSTM models was accordingly pre-processed. After evaluating the model results, it was found that the second study case performed better than the first one, showing the value of the data collected from clients. These findings highlight the project's potential for further development and its capacity to unlock promising possibilities in the field.

Preface

This Master's Thesis means the culmination of my Master studies in Industrial IT and Automation at the University of South-Eastern Norway (USN), campus Porsgrunn. I started these studies in August 2021, after finishing my Bachelor in Industrial Electronics and Automation Engineering at the University of La Laguna (ULL), Spain. This meant the beginning of a new period in my life, leaving almost 22 years of living in the warm Tenerife to move to the cold Porsgrunn, without knowing the language or anyone in the country. But it was for sure one of the best decisions I have made so far. Norway has been revealed to be a very cosy country and Norwegians, very welcoming people.

I want to thank you USN for giving me the possibility to course this Master in their facilities and for allowing me to change from campus to industry master modality in August 2022, where I could work in my field while completing these studies. Within this regard, a special thanks to Håkon Viumdal, who has played a key role in my academic trajectory at USN and who accepted to be my main supervisor in this thesis. A big thank you also to Ru Yan, who also accepted to collaborate in this thesis, and has also been with me as a teacher from the beginning of this adventure.

The second big thanks goes to ENRX AS, or actually to EFD Induction AS (which was the name of the company when I started in it), for allowing me to course the industry master program and this thesis within the company. Here, I must mention and thank Bjørn Sverre, who guided me from the beginning within the company, giving me the possibility to learn and develop my skills under his leadership in the R&D department. He was the one who helped me find the right topic for this thesis and accepted to follow it. Another big appreciation goes to Audun Hystad, a colleague of mine in the R&D department, who did not hesitate to accept my proposal for being the technical supervisor in this project, even if he already had a lot of other projects and tasks to accomplish. I am also very happy to continue my adventure as a full-time employee in this exciting company, now that I have completed my Master's studies.

Last but not least, a big thanks to my family and friends, who have also been there following and caring about the development of this thesis. Here I want to mention my mother, grandmother and uncle who have been supporting me since the first day I came with the idea of moving to Norway, and who have never stopped caring and helping, even from the distance. And in my family, I want also to include a person who I now call my "Norwegian mum", who I met thanks to a passion that we have in common, the music. She has been an enormous help to me since I moved to this country, even in finding the right company for my studies and work. And the last thanks goes to my girlfriend, who has been handling my changes of humour at home during these months, giving me the emotional support needed and always willing to help.

Thank you very much to all of you because this result would not have been possible without your support.

Stefano Fernando Panzeri Reyes.
Skien, 28th October 2023.

Contents

Preface **3**

Contents **5**
List of Figures 6
List of Tables 7
Nomenclature 8

1 Introduction **9**
1.1 Background 10
1.2 Problem Description 11
1.3 Literature Review 12
1.4 Report Structure 14

2 Theory **16**
2.1 Data Science 16
2.2 Machine Learning 17
2.2.1 Neural Networks 19
2.3 Induction Heating 22
2.3.1 Induction Hardening 23

3 System Description **25**
3.1 ENRX Machines 25
3.1.1 Frequency Converter 25
3.1.2 HardLine Machine 26
3.1.3 Customers' Machines 28
3.2 Remote Services System 29
3.2.1 Remote connection 29
3.2.2 Datalogging 30

4 Exploratory Data Analysis **33**
4.1 Evaluation and Filtration of data set's features 34
4.2 Correlation Analysis Development 37
4.3 Common Strong Linear Correlations 39
4.4 Relevant Non-Linear Correlations 41
4.5 Summary 43

5 LSTM Neuronal Network - Error Prediction **44**
5.1 Problem Selection . . . 46
5.2 Data Selection . . . 50
5.3 Data Preparation . . . 52
5.4 Model Development . . . 55
5.5 Model Performance Evaluation . . . 59

6 LSTM Neuronal Network - Improved Error Prediction **63**
6.1 Data Selection . . . 65
6.2 Data Preparation . . . 66
6.3 Model Development . . . 67
6.4 Model Performance Evaluation . . . 69

7 Conclusions **72**
7.1 Future work . . . 74

References **76**

A Master Thesis Description **80**

B Frequency Converter's Nomenclature **83**

C Collected variables from machines **84**

D Evaluation of feature's relevances - Working Table **86**

E Client's Correlation Analysis - Tables and Graphs **89**
E.1 Correlations Client 1 . . . 90
E.2 Correlations Client 2 . . . 97
E.3 Correlations Client 3 . . . 102
E.4 Common Correlations . . . 107

List of Figures

1.1 ENRX main induction heating equipment [4]. 10

2.1 Machine Learning branches and usage [17]. 18
2.2 Neural Networks Architecture [20]. 19
2.3 Multi Layer Perceptron [21]. 20
2.4 Convolutional NN + MLP [21]. 20
2.5 Recurrent NN [21]. 20
2.6 Memory cell structure of an LSTM Network [26]. 21
2.7 Eddy Currents phenomena. 22
2.8 Frequency Converter - Simplified Block Diagram. 23
2.9 Gear tooth hardening [33]. 24

3.1 ENRX Serial Frequency Converter - Block Diagram. 26
3.2 3D Schematic of the Hardline M - Overview. 27
3.3 3D Schematic of the Hardline M - Main body details. 27
3.4 Remote Services System - System Sketch. 30
3.5 Datalogging - System Sketch. 31
3.6 Datalogging system- physical installation on a client machine. 32

4.1 EDA Development - Simplified Block Diagram. 33
4.2 Relevant correlations from Scatter Matrix - Client 3. 42

5.1 LSTM Case 1 Development - Simplified Block Diagram. 45
5.2 Error type distribution for the selected client and job. 49
5.3 Standardization of Data - equation and distribution [41]. 54
5.4 *Sigmoid* activation function [43]. 56
5.5 Confusion Matrix and relevant Performance Metrics. 57
5.6 *ReLU* activation function [43]. 57
5.7 Performance Metrics of Best Model - Study Case 1. 60
5.8 Model Predictions on test data - Study Case 1. 62

6.1 LSTM Case 2 Development - Simplified Block Diagram. 64
6.2 Performance Metrics of Best Model - Study Case 2. 69
6.3 Model Predictions on test data - Study Case 2. 71

List of Tables

3.1 Client's Frequency Converter Technical Data. 29

4.1 Filtered and Adjusted Variables. 37
4.2 Common strong correlations for all the three clients. 40

5.1 Error Type Counting - Client 1. 47
5.2 Error Type Counting - Client 2. 47
5.3 Error Type Counting - Client 3. 48
5.4 Best Models Results - Case 1. 59
5.5 Performance metrics over test data. 62

6.1 Best Models Results - Case 2. 68
6.2 Performance metrics over test data. 71

C.1 Variables Collected. 84

Nomenclature

Symbol	Explanation
AC	Alternate Current
AE	Auto Encoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CSV	Comma Separated Values
DC	Direct Current
EDA	Exploratory Data Analysis
FNN	Feed-forward Neural Network
HMI	Human-Machine Interface
ICA	Independent Component Analysis
IoT	Internet of Things
LAN	Local Area Network
LED	Light-Emitting Diode
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
NN	Neural Network
PCA	Principal Component Analysis
PLS	Partial Least Squares
PLC	Programmable Logic Controller
RNN	Recurrent Neural Network
SCP	Secure Copy
SIM	Subscriber Identity Module
SSH	Secure Shell
SVM	Support Vector Machine
SW	Software
UDP	User Datagram Protocol
WAN	Wide Area Network

1 Introduction

Nowadays more and more industries, especially process industries, are trying to fit themselves in the so-called Industry 4.0. This term refers to the fourth industrial revolution, which is powered by cyber-physical systems. Different technologies are playing a key role in the development of this new type of industry, such as [1], [2]:

- IoT devices and digital twins.
- Cloud connection and cybersecurity.
- Data collection and analysis using AI.
- Human-machine interaction (automation and robotic systems).

Among those points, industries are now focusing especially on data collection and analysis using artificial intelligence methods. This is because these methods can be used to improve process efficiency and productivity, help to make better decisions, improve customer experiences, etc.

One of the biggest challenges when collecting data is to select the appropriate variables to collect as well as their collection frequency. Questions such as: "How often should each variable be collected?", "Is this variable needed?", "Should all the data points be logged or just its average, mean and max?", always arise when setting up a data collection system. In some cases, there is also the possibility that the data available from the system is not enough or does not give all the information needed such as, for example, process failures. All these challenges increase when the process itself has fast dynamics, where changes can happen in fractions of a second.

When having all the data collected, it is important to choose the right methods to analyse it. Artificial Intelligence (AI), and more specifically Machine Learning (ML) algorithms, are innovative and powerful tools to analyse data, but they need to be used and trained in the right way to get meaningful results. This does not only mean choosing the appropriate ML method to analyse the collected data (which is a challenge itself) but it also means providing the proper data to it. This can include, among others, filtering and pre-processing methods of the collected data, which can vary a lot depending on the process under study and on the type and amount of data collected.

In this project, which is done in collaboration with ENRX AS, data collection and analysis from three different induction hardening processes have been accomplished. Three ENRX

customers in different locations that carry out induction hardening on several different workpieces, agreed to share process data. To do so, an installation was done in the clients' machines to collect this data remotely and analyse it afterwards. The goal of this study is to work with this data and see if some meaningful information can be extracted from it. The idea is to focus both on the clients' hardening processes and the induction heating machine delivered by ENRX. The result can lead to improved working efficiency or the implementation of a predictive maintenance system in the future.

1.1 Background

ENRX is an international green technology company that delivers equipment and solutions related to induction heating, wireless inductive charging and contactless power supply. It was born in March 2023 by the fusion of EFD Induction and IPT Technology, two well-established companies in the induction heating and inductive charging market respectively [3].

Within the induction heating market, ENRX provides a variety of products with different useful applications. The main products regarding induction heating equipment/machines, which are shown in Figure 1.1, are (from left to right): Minac (mobile induction heating equipment), Weldac (tube welders), Sinac (stationary induction heating equipment) and HardLine (induction hardening machines) [4]. Some examples of the numerous induction heating applications are bolt heating, bonding, hardening, bracing, welding, tempering and shrink fitting. [5].

Figure 1.1: ENRX main induction heating equipment [4].

ENRX is not a process company itself, but the machines delivered can be installed in the customer's production line. Therefore, the delivered machines are already equipped with the necessary technology to integrate themselves into Industry 4.0. ENRX wants now to

go a step forward within that area and therefore it has an ongoing IoT project related to machine connectivity and data collection and analysis.

The named project consists of two main parts which are called: Remote Services and Information Services. The first part is focused on machine connectivity over the internet. As named before, ENRX is an international company that has over 25,000 induction heating machines installed in more than 80 countries around the world. If a customer has a problem with a machine, the ENRX service department tries to solve the problem by email or phone first but may lack of real-time overview of the machine status or problems. In such cases, the solution could just be a software fix, an update installation or a change of configuration in the control system, but this is very hard to know if the service engineer is not able to see the machine status. Therefore, this could result in a service engineer travelling physically to the customer to solve the problem. In some cases, this can be seen by the customer as a waste of time and money, not only because of the possible ENRX service cost but also because of the downtime of its production line. Therefore, the first part of this project is focused on developing a system that will provide remote access to the control system of the delivered machines, so that a service engineer would be able to provide online support to overcome issues that have arisen. Nowadays only a limited number of systems delivered can be remotely connected, and the process of doing so needs to be streamlined.

Taking into consideration that the machines will be connected to the internet and the information from the control system will be available, the second thought is to start collecting and analysing the functioning data of the machines. The involvement of ENRX in Data Collection and Analysis, which is a trend topic within Industry 4.0, will give the possibility to the company to create a whole new business case with a new service to offer to its customers. As explained before, the collected data will hopefully bring some meaningful information to the customer about its process and to ENRX about its machines. Before setting up a whole cloud collection system, it is important to know which data to collect and if the information that is possible to extract from it is valuable. Therefore, this project could be described as the first technical feasibility study of the Information Services system, hoping that it could bring some meaningful information and guidelines on how to continue with that part of the project.

1.2 Problem Description

As stated in the above sections, the goal of this study is to try to solve a Data Analysis problem to gain some meaningful insight into the information that can be collected from ENRX's machines. To do that, a large amount of data from different process variables needs to be collected and analysed using different methods to extract value from it. Therefore, in this document a collection and analysis system will be built and the results

obtained from this analysis will be interpreted and discussed. The aim is to extract information that can be used by ENRX to create a new business model that will improve its customer's services as well as its products.

To obtain real process data, three customers from ENRX have agreed to collaborate by sharing data from their machines. As explained before, ENRX's induction heating machines differ both in characteristics and functionalities, so for a more accurate comparison study, three customers with the same machine type and with the same application usage have been selected. The three of them carry out induction hardening on different pieces using the HardLine machine. In addition, all three of the customers had already installed in their machines a prototype system for Remote Services. This means that the machines were already remotely accessible, having therefore the possibility to implement a data collection system.

A set of process variables, as well as other relevant informative variables are collected every time a new piece is hardened. Even if the overall process is the same for every workpiece and every customer, the hardening result wanted in each one of them differs. Therefore, a group of settings will be loaded in the machine for each different workpiece, to transfer the correct amount of energy to it. This means that, theoretically, two data sets from two different workpieces can contain similar data, but have a completely different meaning. The same amount of energy transferred to two different pieces could mean that one piece is correctly hardened while the other one is not. In addition, the collected data set does not contain the specifications of the hardened workpiece, but just the configuration loaded in the machine for that specific hardening process. Therefore, in this study, it will be assumed that a specific configuration (also called, job/recipe) will always be used to harden the same type of workpiece.

It is observable that just with one machine (HardLine) in use for one application (induction hardening), a lot of singularities already exist. One of the most challenging aspects of this study is to generalize the results obtained for a general induction hardening process or a general induction heating process.

1.3 Literature Review

A large number of papers can be found on topics related to Data Analysis, Data Mining, Machine Learning, AI Process Modelling, etc., but it is quite hard to find papers that apply these techniques to induction hardening processes. The topic of this study is very specific, therefore, for the literature review, papers discussing related topics or methods applied to similar processes have been read to extract some meaningful information that can be valuable for the problem under study.

In this regard, a paper titled *Data Mining and Analytics in the Process Industry: The Role of Machine Learning* [6] has been found very interesting for this study. In this paper, different general data mining topics are explained in detail and discussions on how they are used or applied in the Process Industry are found. The paper covers the four main steps needed to apply Data Mining and Analytics, which are: Data Preparation; Data Pre-processing; Model Selection, Training and Performance Evaluation. In these three steps, a quite detailed overview is shown, explaining how process data should be analyzed according to its nature, the importance of identifying outliers and scaling/transforming the data before using it in a model, how to select the right model to train according to the data type and the expected outcome, and how to validate the accuracy and usefulness of a model in the real world.

In the third section of the paper, the main Machine Learning methods used in process industries are described, being: unsupervised learning, supervised learning, semi-supervised learning, and reinforcement learning. Within these four main groups, many different methods are described in detail and their applications are pointed out. It is also noted that 80 % to 90 % of all the industry applications usually apply supervised or unsupervised learning. Within supervised learning the three main methods used are: Artificial Neural Networks (ANN), Partial Least Squares regression (PLS) and Support Vector Machine (SVM). On the other hand, within unsupervised learning methods, the three main sub-methods used are: Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Manifold Learning.

The content of this paper could be used as a main step guide when preparing the data, deciding and training the model and analysing the results. Many good explanations and examples are given that can be applied to the problem under study, due to that the data collected also comes from processes in industries.

Looking into research papers that are more focused on the field of study, the paper named *'Data Mining Methods - Application in Metallurgy'* [7] is of relevance for this study. In this paper, different data mining methods are described with a focus on their application in the metallurgy sector. In this regard, examples of data analysis, and modelling and control of metallurgical processes, such as the copper flash smelting process are presented. After explaining with text, graphs and equations the functioning principles of methods like K-means clustering, decision trees, or artificial neuronal networks, some of them are applied with different purposes in different metallurgical processes.

One of the most interesting aspects of this paper is the section where the identification of the copper flash smelting process is carried out. In this process, different methods such as ANN, decision tree, non-linear regression and linear regression are used to predict the NO_x concentration in the gases emitted. The results point out that for this process, ANN and decision trees obtain the best results among the four of them.

The content of this paper can be of great use when needing to understand the working

principle of different methods that can be used to analyse and model the process under study. In addition, induction hardening is also considered as a metallurgical process, therefore, some of the results presented in this paper can be of use for inspiration or comparison with the study case.

In this project, the data set will contain time-series data, and therefore it is interesting to have a look into more specific methods that can be used to analyse this type of data set. In this regard, a paper which can be a very good starting point for the development of this project is the one titled *'LSTM-Based Stacked Autoencoders for Early Anomaly Detection in Induction Heating Systems'* [8]. This paper discusses the issue of accurately measuring the temperature of cookware on induction heating systems when the cookware isn't centred properly on the heating coil. To do that, the study introduces a data-driven anomaly detection method that uses thermal sensors to detect overheated cookware by monitoring the inverter module's case temperature. This method uses a long short-term memory (LSTM) based autoencoder (AE) trained on extensive data of cookware and inverter temperatures, obtaining a model that can detect overheating and inverter faults. The study also compares the LSTM-AE with other deep learning algorithms and finds it to be more effective in detecting anomalies.

Many other papers and/or books can be found about related topics, but to conclude this chapter, a last document will be named. This paper, titled *'Anticipating Future Behavior of an Industrial Press Using LSTM Networks'* [9], introduces predictive models using LSTM neural networks. To do that, these methods are applied to sensor data from an industrial paper press over three years. After pre-processing and optimizing the collected data, the models can forecast equipment statuses up to one month in advance with reasonable confidence. These predictions enable better maintenance planning and further research to enhance model reliability. The study case of this paper is very similar to the results that want to be achieved in this project, therefore it can be interesting to look at this type of algorithms for the development of this study.

The information from the papers cited in this section, together with the citations that will appear in the next sections of this document will be used as a help and reference for the work carried out in this project.

1.4 Report Structure

This report will not strictly follow the standard IMRaD structure, but it will be structured by the most relevant topics (which will contain the used methods, results and discussions) in addition to some general chapters. This structure is intended to make it more understandable and easier to read. Therefore, after this introduction chapter, a chapter containing the basic theory that needs to be understood to read the report is presented. After that, the system description will be found, with a focus both on the

machines under study and the system used to collect the data from them. Then, the main topics of this study will be presented, which are Exploratory Data Analysis (EDA) and LSTM Neural Networks. These two methods have been used to analyse the data collected and extract conclusions from it. The Exploratory Data Analysis has been primarily used to analyse linear and non-linear correlations between variables in the data set, while LSTM Neuronal Networks have been used for error prediction with two different types of input data. These chapters will present the work done on these topics, as well as the results obtained with appropriate discussions. The report will finalize with a conclusion chapter, that will summarize the aims achieved and the knowledge gained in this project, pointing also to future lines to research. Relevant appendices will be found at the end of the report.

Within this project, several Python scripts have been developed to handle and analyse the collected data. All the code, as well as some other relevant files that have been used, can be found in a GitLab Repository [10] with the same name as this project. Due to privacy reasons, all the collected client data cannot be uploaded to a public repository, as well as some of the results obtained. Therefore, many of the Python scripts which need as input the folders containing the collected data, will not work due to lack of references. To give the possibility to at least be able to execute the LSTM NN, two Jupyter Notebooks for each study case will be found. One of them will contain the whole LSTM development process, from data selection and preparation to model development and evaluation. This script will save the vectors used as input for the development of the ML algorithm to local variables. The other one will only have the training and evaluation part of the LSTM development, using as input the variables previously saved. This will give the possibility to the reader to test these algorithms itself. In each section of this report where some of the Python scripts present in the repository were used, a reference (with a clickable link) to its path and/or name will be given in the corresponding section. All the Python files, Jupyter Notebooks or folders that contain them will start their names with an "S" (which stands for section), followed by the section number and an underscore, followed by the subsection number the file corresponds to. After that, two other underscores with another number will indicate the file number within that section/subsection. The name will be finalised with an underscore followed by the script name. If a file belongs to several sections of the report, this will start the same way as described, and the different sections will be separated by a double underscore.

For example: If a Python file with the title "ScriptName" is present in Section 4.1 (where 3 Python files are referenced), and this is the second file that appears in the text, this will have the following name: *S4_1___2_ScriptName*. If a Jupyter Notebook named "NotebookName" contains the information from sections 4.1 and 4.2. this will have the following name: *S4_1___S4_2_NotebookName*. This will make it easier for the reader to associate each file with the corresponding section of the report and to understand in which order they have been executed for the whole project development.

2 Theory

In this chapter, an overview of the theory behind the main elements of the project is given. This includes both definitions of concepts related to Data Science and Data Analysis, as well as Induction Heating and Hardening, which correspond to the main functionalities of the machine under analysis.

2.1 Data Science

Data Analytics is a set of scientific methods, which are focused on analyzing data sets to extract meaningful information from them. Trying to understand data has been addressed throughout history by many statisticians, mathematicians and scientists among others. However, it was John Tukey who established the term Data Analysis in 1962. He was the one who pointed out the important difference between **Exploratory Data Analysis**, which refers to representing the data visually and obtaining a summary out of it, and **Confirmatory Data Analysis**, which refers to statistical analyses driven by rigid mathematical methods. He also articulated the importance of using computer science and graphs, especially in the Exploratory Data Analysis. The combination of Data Analytics and Computer Science is what is known nowadays as Data Science, which combines the application of statistical, computational, and machine learning methods to extract insights from data and make predictions [11], [12].

Raw data is sometimes not that useful for analysis, therefore this is usually manipulated and transformed using different methods to visualize patterns or extract the information of interest. Data Analytics can be group in 4 different types ([13]) that answer to the corresponding questions:

- Descriptive Analytics: “What happened?”
- Diagnostic Analytics:“Why did that happened?”
- Predictive Analytics: “What will happen in the near future?”
- Prescriptive Analytics: “What needs to be done?”

To answer the above questions, various techniques such as regression analysis, time-series analysis or Monte Carlo simulations can be used. These techniques were first applied by skilled mathematicians but over the years, more and more software tools have become available to optimize these studies. Nowadays, because of the large amount of data that is usually needed to handle by data scientists, programming languages (especially Python) with ML algorithms implemented are becoming more and more popular to analyse data.

2.2 Machine Learning

Machine Learning is a subset of Artificial Intelligence that is focused on analysing large amounts of data using different algorithms with the purpose of learning patterns and relationships in data sets. These algorithms build mathematical models by constantly analyzing and processing several sets of data to make predictions on a new unseen data set. These algorithms do not rely on explicit instructions, but they learn (in a similar way as humans do) how to identify patterns, extract meaningful features, and make informed predictions or decisions.

Machine Learning Techniques are widely used nowadays in many different sectors and for many different tasks such as social media optimization, financial prediction, product recommendation, etc. According to the type of problem to solve or to the available data set, different Machine Learning techniques can be used [6], [14], [15]. The most popular ones are now described together with their main applications.

- Supervised Learning: this category is designed to build mathematical models based on a set of data which is labelled. This means that the data set has defined inputs which are associated with a known target or output. These combinations allow the model to learn over time, measuring its accuracy in every iteration and adjusting until the error has been sufficiently minimized. This method can be used both for classification, which is intended to classify the data into different categories; and for regression, which is intended to find dependencies between variables and make projections. Some common usage examples are voice and facial recognition and spam email detection.
- Unsupervised Learning: this group of methods is intended to be used with sets of data that are unlabeled. The model is designed to find structures and patterns to group the data set in clusters. Those clusters help identify similarities and differences in a data set and identify those in each new piece of data. This method is very relevant for applications like density estimation, anomaly detection, customer segmentation, or medical imaging, among others. In addition, unsupervised learning can also be used for dimensionality reduction, which consists of reducing the number of features/variables under consideration in a given data set. By doing

that, the analysis becomes less complicated, overcoming the "curse of dimensionality" that arises when analysing data in high-dimensional spaces [16]. By doing that, unsupervised learning becomes very useful in analysing Big Data, where many different variables or dimensions are present.

- Reinforcement Learning: this type of ML uses trial-error methods, which do not rely on an exact mathematical model. These methods have three main elements which are: the environment, the interpreter and the agent. The agent takes some actions over the environment and the interpreter determines if the outcomes of these actions over the environment are the desired or not. If that's the case, then the interpreter reinforces/rewards the agent and vice-versa. The agent will take action to maximize the cumulative reward received over time, which could cause a change in the environment's state. These types of algorithms are ideal for making decisions in uncertain environments, such as in autonomous vehicles or game simulators (where the machine needs to compete against the human).

In Figure 2.1, a summary of the three main Machine Learning branches and their main usages are shown.

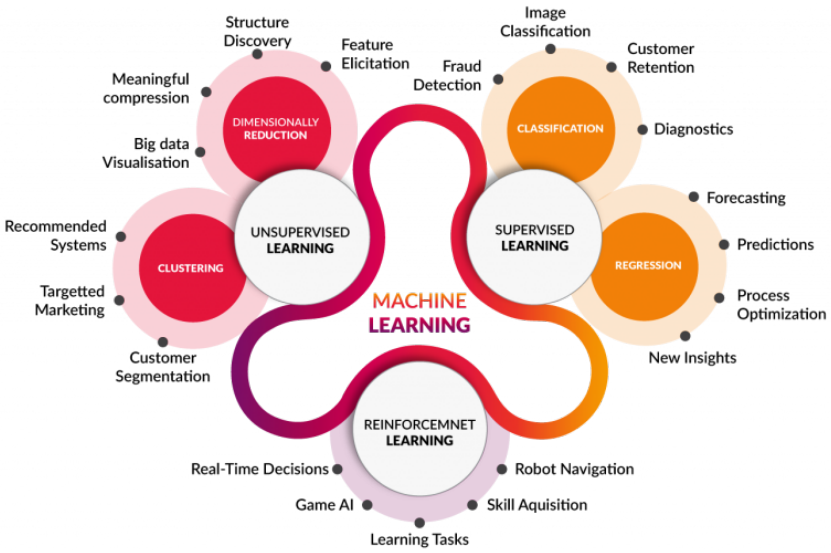


Figure 2.1: Machine Learning branches and usage [17].

Machine Learning capabilities to learn and extract information from data in a very efficient and accurate way makes it very suitable and a main analysis technique in Data Science. The information that these algorithms can extract from a data set, can be the keys that drive decision-making within applications and businesses. With an increase in big data in recent years, the demand for data scientist who can use ML methods as tools in their analysis is growing exponentially.

2.2.1 Neural Networks

Neural Networks (NNs), also called Artificial Neural Networks (ANNs), are one of the most commonly used subsets of Machine Learning. ANNs consist of a connection of different artificial neurons that try to model the connection and behaviour of human neurons. These networks are structured in different layers containing one or several of these artificial neurons (also called nodes). All neural networks must at least have an input and an output layer. Depending on the complexity of it, they can also contain one or more hidden layers. Each node is connected with one or several nodes of the next layer with an associated weight and threshold. Each input received in that node is multiplied by its corresponding weight and the sum of them corresponds to the output of that node. Only if this output is above the specified threshold, then the node is activated and the data is sent to the next layer. Neuronal Networks that contain more than 3 layers in total, are considered as Deep Learning Neuronal Networks [18], [19]. NNs are a building block for ML and therefore can be used for any of the three groups described in the previous section.

In Figure 2.2, a general fully connected neuronal network structure is shown.

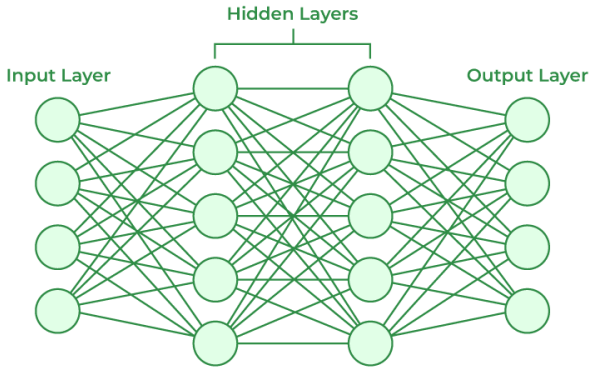


Figure 2.2: Neural Networks Architecture [20].

The three most used Neuronal Networks are the following:

- Feedforward NN (FNN): These are one of the most basic types of neural networks. The information travels in only one direction from one or several input neurons to the output node, having static weights. Even if it usually contains one or more hidden layers (being then also called Multi-Layer Perceptron (MLP)), it may not have them. Sometimes, instead of being used alone, a series of Feedforward NNs are used together with a minor intermediary for different applications such as computer vision or pattern recognition [18], [21], [22].

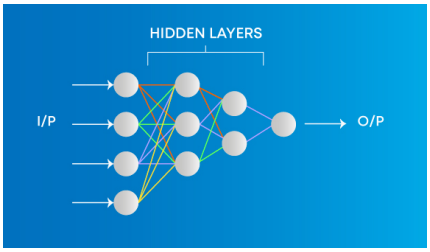


Figure 2.3: Multi Layer Perceptron [21].

- Convolutional NN (CNN): The main characteristic of these NN is that they are specifically designed to process pixel data and are used in image recognition and processing. The first layer of this type of NN is the convolutional layer, which contains several neurons, each of them in charge of analysing a small part of the visual field. To understand the full image, these operations (convolutions) must be repeated multiple times. Usually, the output of a CNN is used as input in a MLP to achieve, for example, image classification [18], [21], [22].

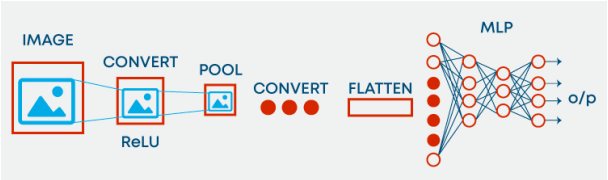


Figure 2.4: Convolutional NN + MLP [21].

- Recurrent NN (RNN): The main element of these NN is that they contain feedback loops in the middle layers. The result of each node in the middle layers is saved and fed back to the input of that node in the next iteration. This will gradually increase the probability of making the right prediction, due to that previous information is taken into account in each one of the nodes. These learning algorithms are mainly used when analysing time-series data to make predictions about future outcomes [18], [21], [22].

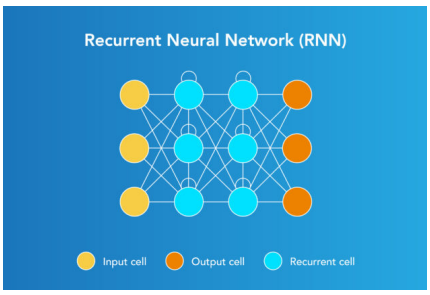


Figure 2.5: Recurrent NN [21].

- LSTM NN: LSTM, which stands for Long Short-Term Memory, is a type of Recurrent Neuronal Network which can retain and discard the relevant information in a long-term prediction. This is the main difference and feature that makes them superior to standard RNNs. LSTM have a memory cell which contains two states: cell state and hidden state. The hidden state is the one that is also present in standard RNNs, which retains the information of the immediately previous events. On the other hand, the cell state describes the long-term memory capability of LSTM, which allocates relevant information from previous events that do not necessarily have happened immediately before. To decide which information needs to be kept, removed or output from the memory cell, three gates are used to control it: input gate, forget gate and output gate. Each one of these gates is built using two elements a *Sigmoid* neuronal net layer and a point-wise multiplication. The sigmoid layer outputs a value between 0 and 1, indicating how much of each component will go through to the next block. The forget and output gate have also a *tanh* layer in addition, which provides a vector of possible candidates to evaluate, which are then filtered by the sigmoid function [23]–[25]. The structure of the memory cell of an LSTM Network is presented in Figure 2.6.

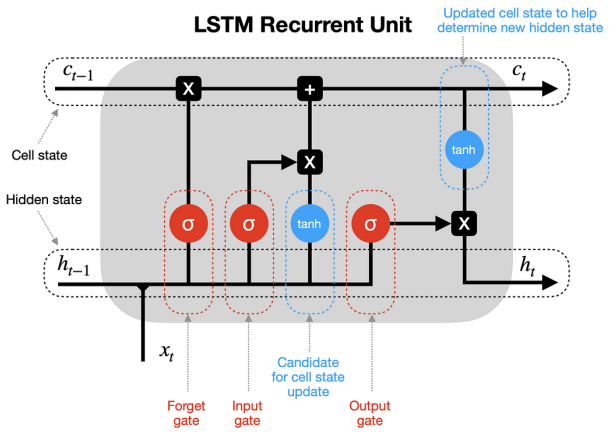


Figure 2.6: Memory cell structure of an LSTM Network [26].

Many other different types of Neuronal Networks with different degrees of complexity exist, such as Radial Basis Functional Neural Networks, Modular Neural Networks, Sequence to Sequence Models, etc. The right one should be chosen according to the problem to solve, to reach a balance between complexity and computational time needed [21], [22].

2.3 Induction Heating

Induction heating is a fast, efficient, flame-free and no-contact heating method used on metals and other conductive materials. To achieve the heating of the desired piece, this is inserted inside a coil that conducts a certain amount of current. By alternating the current flowing through this coil, a magnetic field is generated and eddy currents are induced in the piece to heat. These are close-loop currents that flow in perpendicular planes of the magnetic field (as shown in Figure 2.7) and behave according to Faraday's Law of Induction. This law describes the electromotive force produced in a circuit when this is under the influence of a fluctuating magnetic field, as stated in Equation 2.1.[27]–[29].

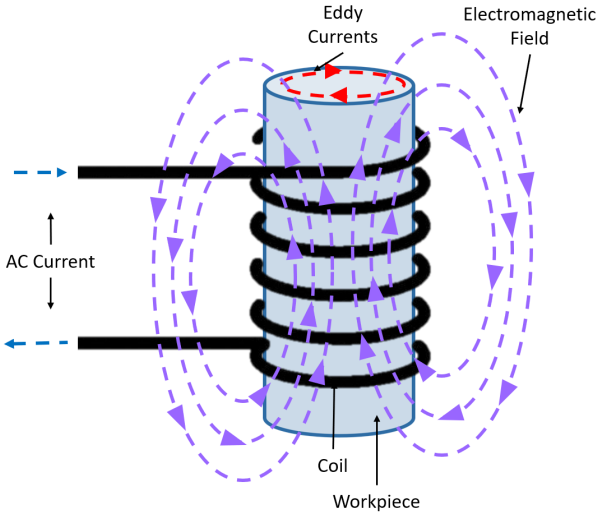


Figure 2.7: Eddy Currents phenomena.

$$EMF = N\frac{\Delta\phi}{\Delta t} \tag{2.1}$$

where:

EMF = Electromagnetic force (V)
N = Number of turns or loops of the coil
$\Delta\phi$ = Change in the magnetic flux ($V \cdot s$)
Δt = Change in time (s)

The heat induced to the object is obtained due to the electrical resistance present in the area of the object where the eddy currents appear. Due to that these currents depend on the magnetic field generated by the coil, a change in this will imply a change in the heating effect. In addition, the heating effect is also dependent on the number of turns in the coil, on the distance between the object and the coil and on the magnetic properties of the workpiece and area around the coil.

To generate this type of heat, frequency converters are designed according to the desired specifications. These machines convert the AC current provided by the mains (which is at 50 Hz or 60 Hz) into a high-frequency current suitable to perform the desired job. To generate this output, the input voltage is converted into a DC voltage, which is then fed through a transistor inverter bridge to generate a high-frequency AC current in the output coil. The intensity of the output AC current will be directly related to the magnetic field generated in the coil, while the frequency of it will influence the heating depth in the material. A lower frequency will heat deeper in the object while a higher frequency will result in a more superficial heating. A simple sketch illustrating the working principle of a frequency converter is shown in Figure 2.8.

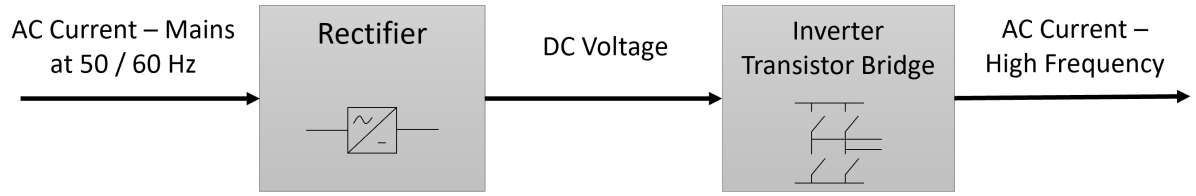


Figure 2.8: Frequency Converter - Simplified Block Diagram.

Induction heating is a very efficient heating method, being able to reach high temperatures in short periods, with small energy losses to the ambient. It is a contactless and precise method, where no flames or gases are produced and only the desired area of the object is heated. Another important characteristic of this method is its high repeatability, which gives the possibility to obtain the same result over the same type of workpiece. [27].

2.3.1 Induction Hardening

The hardening process is a heating process that is used to increase the strength of metal pieces. To achieve that, the object is heated above its critical temperature to obtain a molecular transformation. Then, the object is quickly cooled down to “lock” the mentioned transformation. The result is a harder surface with a higher resistance to plastic deformation.

Using induction heating to achieve this goal is an efficient solution and the one that provides superior results in many cases. As explained in the above chapter, the material can reach high temperatures in very short periods and only in the desired area, obtaining a very accurate and repeatable hardening result. This is why induction heating is often selected to harden gear teeth (as shown in Figure 2.9), piston rods and rotating shafts, where precision is crucial [30]–[32].

Figure 2.9: Gear tooth hardening [33].

3 System Description

To understand the working principle of the data collection system that has been implemented, it is important to know how the different elements of the system work. Therefore, a basic explanation of ENRX's machines as well as how the Remote Services system works is now shown.

3.1 ENRX Machines

As explained in Section 1.1, the range of products delivered by ENRX is large. For this project, the main focus will be to understand the basic functioning of a standard frequency converter and a Hardline machine. Once this is done, an overview of the technical specifications regarding the specific client's machines used for this study will be also given.

3.1.1 Frequency Converter

To meet the different industries' applications, ENRX provides frequency converters in different power and frequency ranges. In the design process of these machines, one of the most important things is to meet the desired output current and frequency for the piece to heat. This is obtained by modifying the output circuit of the converter, composed of a transformer, a capacitor and an inductance (the coil itself). Therefore, each frequency converter is designed for the customer's needs and only works optimally within a specified range of operations.

ENRX also designs a common control system for all its machines. This system is composed of a set of different electronic cards which monitor the converter in real time, keep the output of it in a safe operating area, and act against failures. This control system also interacts with the outer world with different communications protocols and with the operators through an operating panel.

Induction heating processes cover a big temperature range, from temperatures that could be as low as 100ºC to temperatures above the workpiece's melting point. During the energy transformation that occurs inside the equipment, power losses are present and

these are dissipated as heat. Therefore, almost all of ENRX's induction heating equipment (including coils) are water-cooled. In Figure 3.1 it is possible to observe the main components and the working principle of ENRX's induction heating equipment.

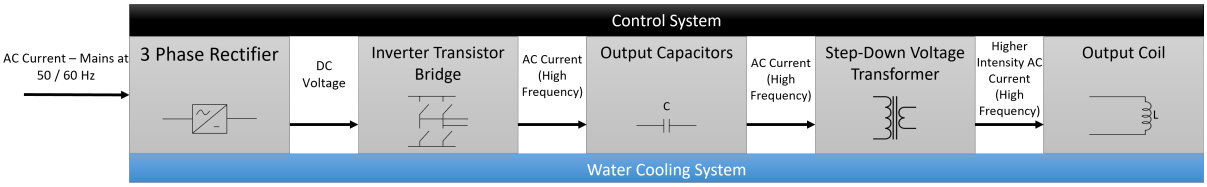


Figure 3.1: ENRX Serial Frequency Converter - Block Diagram.

3.1.2 HardLine Machine

The Hardline machine is a modular induction hardening machine that is built with different modules/blocks according to the customer's requirements. The two main elements that are always present in this machine are a Sinac frequency converter and an axis machine controlled by a PLC. The converter supplies the necessary power to the coil, while the axis machine is the one moving the coil along the piece and controlling the quenching fluid to harden the piece.

The axis machine provides movement in the X, Y, Z and C axes, which correspond to right/left, back/forth, up/down and rotational movement respectively. The most common movement combination is the one where the piece to harden is rotating while the coil is moving upwards to heat the piece along its length. The rotation ensures a uniform heating, that could be affected by the coil shape or a piece not placed in the exact center of it. In the data collected for this project, some customers also make use of the Y axis, which is mainly used with open coils that need to "get in and out" a part of the piece to harden, as previously shown in Figure 2.9. The piece under hardening is held within the tailstock, which allows the loading of workpieces with different lengths. Hardline machines give also the possibility to insert several pieces at the same time on a rotational table, where, for example, one piece can be hardened, while another is on tempering and another one is finishing its cooling process. A general 3D overview of the machine is presented in Figure 3.2 while a more detailed 3D schematic of the machine is presented in Figure 3.3.

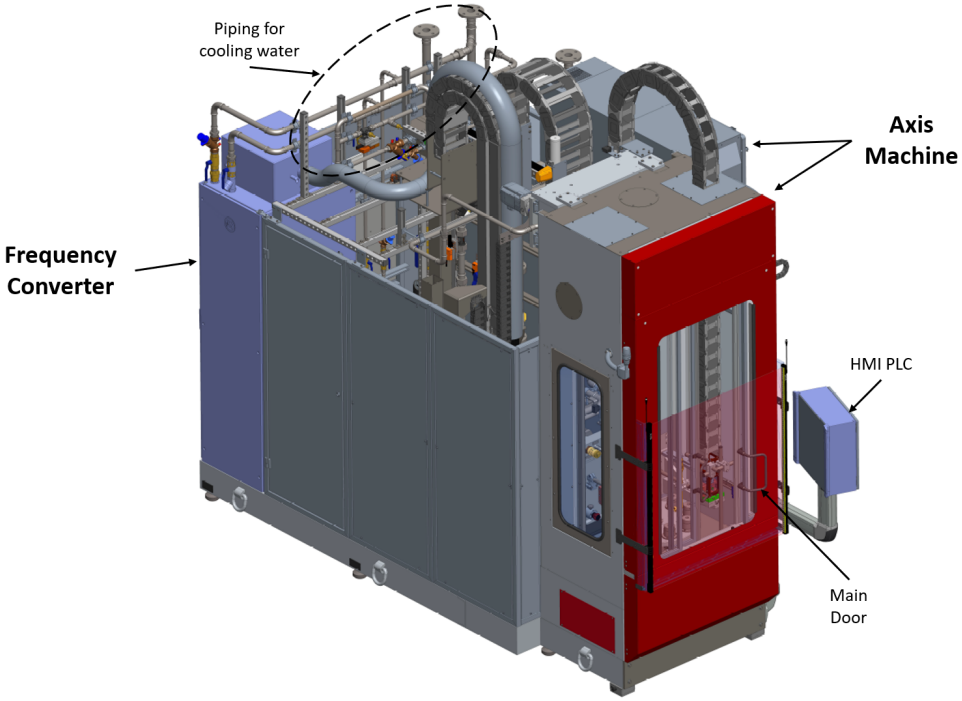


Figure 3.2: 3D Schematic of the Hardline M - Overview.

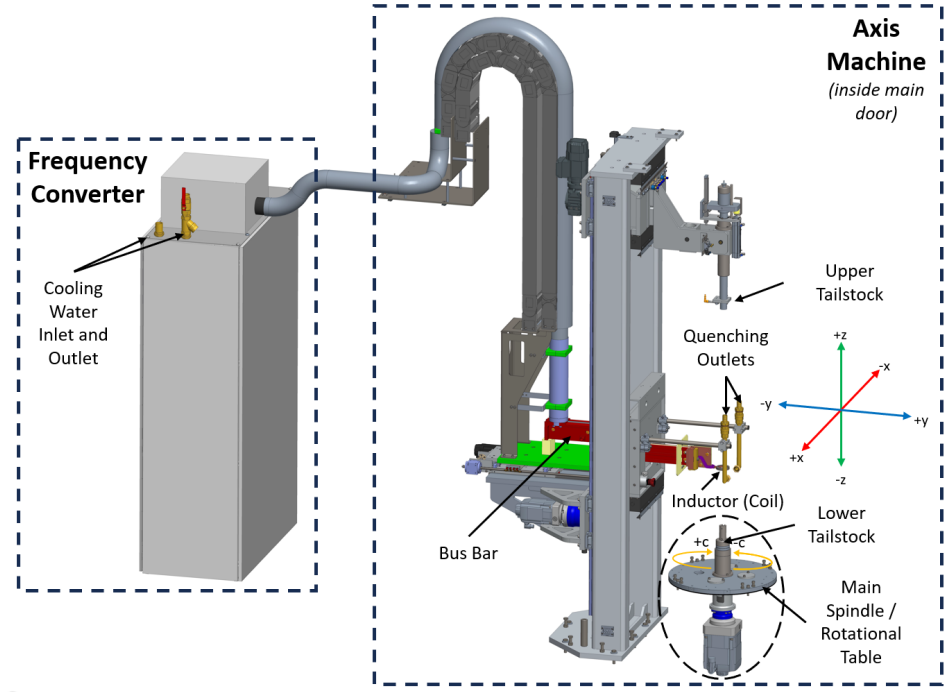


Figure 3.3: 3D Schematic of the Hardline M - Main body details.

Both elements present in the HardLine (the axis machine and the frequency converter), are controlled from the PLC's HMI. In this HMI, the operator selects a job/recipe to execute according to the piece to harden. This job contains a series of pre-set parameters for both the frequency converter and the PLC. Once the process is started, the PLC is the main element in this control system, which is in charge of sending the desired values to the control system in the frequency converter, reading the feedback from it, controlling the axis and the quenching fluid, handling alarms and warnings, etc.

To understand better the working principle of this machine, it is possible to watch this explanation video: HardLine M - Modular induction hardening machine [34].

3.1.3 Customers' Machines

Three ENRX's clients have agreed to share data from their machines for this study. Even if all the clients have a Hardline machine, the frequency converter's specifications of each one of them can differ, due that these are customized. An overview of the frequency converters that are used for this study is now shown. Due to data protection reasons, the name or location of these clients cannot be stated in this report.

Even if the majority of the converters sold by ENRX have an output circuit where the capacitor and the inductance are connected in serial, some of them can have an output circuit where these two elements are mounted in parallel. Of the three clients, one of them has a parallel output frequency converter. This will result in some differences in the data collected, due that in these types of converters, the variable that is controlled, is not directly the output current, but the output voltage of the circuit. One of the most noticeable differences is when the output power of the converter is off (not heating). In serial converters, a DC voltage value will always be present when the power is on and off, but this phenomenon does not apply to parallel converters, where the voltage is only present if the converter is heating (output power is on).

In Table 3.1, the most relevant technical specifications of the customer's machines are summarized. All three Sinacs have a 3-phase input voltage supply between 400 V and 480 V at 50 Hz.

Table 3.1: Client's Frequency Converter Technical Data.[1]

	Client 1 - Sinac 100/160 SM	**Client 2 - Sinac 150/70 SMH**	**Client 3 - Sinac 100 PM**
Max. continuous output power	100 kW	100 kW	100 kW
Max intermittent output power	160 kW	150 kW (MF) / 70 kW (HF)	-
DC voltage nominal	540 V	540 V	510 V
Frequency range	7.5-15 kHz / 30-40 kHz	7.5-10 kHz / 100-150 kHz	3.8-10 kHz
Max. inverter output current	1100 A	900 A	1100 A
Cooling water consumption	54 l/min	42 l/min	20 l/min
Max. inlet water temperature	35 °C	35 °C	35 °C

In addition to the differences in the frequency converters, the whole machine itself has some other differences. The machine from Client 1 uses an external robot to place and remove the pieces to harden in the machine. This robot is controlled by another system and its values are not shared with the system under study. The machine from client 2 has an external quenching station, and therefore the information from it is not available in the actual system.

3.2 Remote Services System

This section will clarify the system used to have a remote connection with the machines under study as well as the system designed and implemented to collect the data of interest from these machines.

3.2.1 Remote connection

The three selected customers for this project had previously installed a prototype of ENRX's Remote Connection System, which allows access and control of the machine and

[1]To understand the meaning of the numbers and letters present in the machines' names, please refer to Appendix B

the devices connected to the same network.

In a very simple way, to gain remote access to the machine, a VPN tunnel between a PC and the router installed in the machine is established. This router has some LAN ports, where different devices can communicate with each other in a local network. In addition, it has a WAN port and a SIM card slot, being able to be connected to the internet through the customer's network or the mobile network. Both the router and the local/operator PC connect to a designated server which is in charge of merging these two connections into one. Once this is done, both devices will be on the same network, being able to communicate with each other. The local PC will have access not only to the router itself but also to the devices connected to the different LAN ports. A simplified sketch to illustrate the working principle of the system is shown in Figure 3.4.

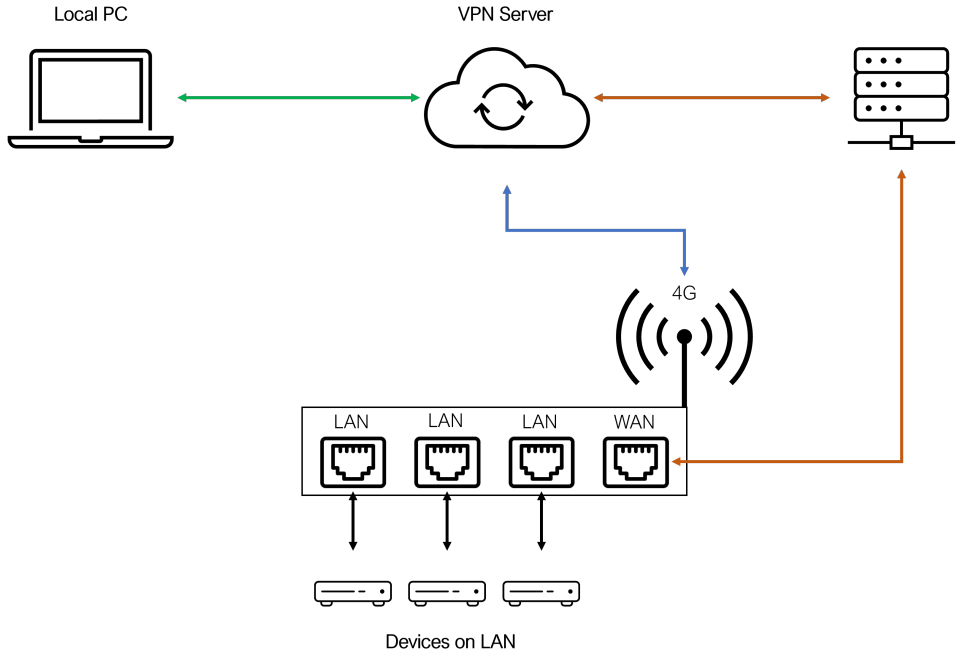


Figure 3.4: Remote Services System - System Sketch.

3.2.2 Datalogging

Even if the system in place gave access to the machine, the data from it was not being collected. Therefore, as a part of this project, a data logging system has been designed and physically installed in each one of the three customers's machines which are part of this study. The design and development process involved the selection of the variables of interest from the system, the development of a Python program to collect the data and the set-up and physical installation and connection of a Raspberry Pi in the present remote

connection system. For the system to work, a small change in the PLC program was also introduced. This was done by an ENRX engineer with knowledge of the system by adding an extra block to the in-place program and not modifying its main functionality, minimizing the risks of introducing a bug.

The designed system has three main elements: a PLC, a Raspberry Pi, and a Router. For this study, the collection system has not been implemented on the cloud, but locally, using the Raspberry Pi as the receiver and storage device. The idea of having a Raspberry Pi (external data logger) instead of accessing the PLC directly is to make the connection more secure, avoiding interfering with the functioning of the machine itself.

The PLC is collecting process and some other relevant variables while the hardening process is ongoing every 200 ms. The variables are packed into a UDP message that is sent to the Raspberry Pi. The Raspberry Pi has a Python program running continuously which is reading, decoding and storing the received UDP message. When the hardening process for one piece is completed, the Python program will create a CSV file with all the data points collected from that hardening process. This file will be saved with the following folder structure */month/day/timeStamp_jobName.csv*. A sketch of the system is shown in Figure 3.5 while a real picture of the physical installation performed in one of the clients is shown in Figure 3.6

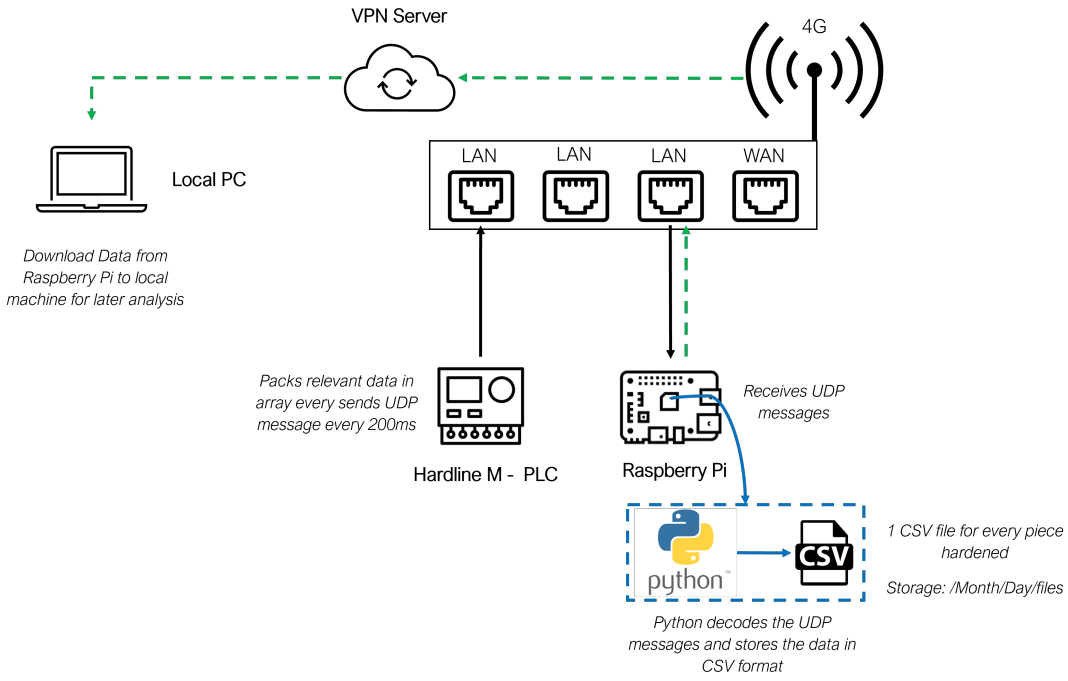


Figure 3.5: Datalogging - System Sketch.

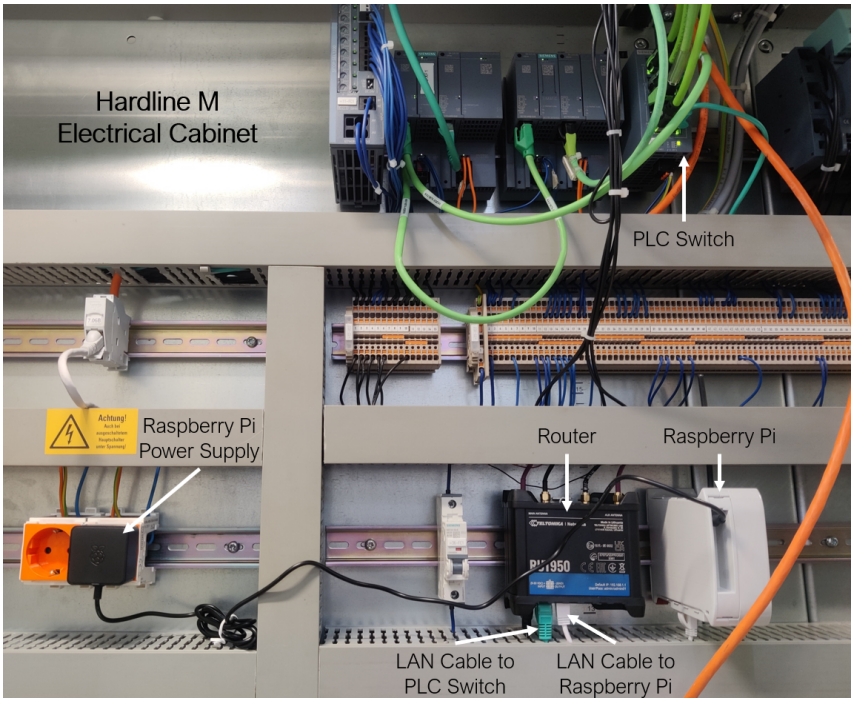


Figure 3.6: Datalogging system- physical installation on a client machine.

Periodically, the Raspberry Pi will be remotely accessed and the files will be backed up to the local PC, for later analysis. To do that, the remote device will be accessed using the SSH protocol and the actual month folder will be packed and compressed into a *.tar.gz* file. This file will then be transferred to the local machine using the SCP command from a local terminal and unpacked accordingly.

The Python code used to collect the data in the Raspberry Pi as well as a functioning description of it can be found under the folder named *S3_2_2_DataCollection*.

An overview of the collected variables is presented in Table C.1 of Appendix C.

4 Exploratory Data Analysis

After collecting process data from April to July from three different customers, this needs to be interpreted. To do that, the first step is to carry out an Exploratory Data Analysis, which can give an overview of how the data behaves to take further actions. To achieve this purpose a very useful and powerful Python library has been used, *ydata-profiling* [35]. This tool can provide a report from a data set indicating correlations between variables, interactions, statistical information from each variable, alerts about the data set structure, etc.

The first step has been to convert all the .csv files collected from the customers into a more manageable and understandable format to apply the analysis. For this reason, a Python program has been used to convert all the CSV files into Pandas DataFrames. Each DataFrame, which corresponds to one CSV file and therefore to one piece hardened, has then been grouped into a dictionary, containing each one as an entry. In addition, a general DataFrame for each customer, containing a concatenation of all the smaller DataFrames, has also been created. These files have been then saved locally in a .pickle format for later use within the library mentioned before. This first conversion is done using the Python script called *S4_0_csv2pickle.py*.

A simple overview of the development of the EDA is presented in the block diagram form Figure 4.1.

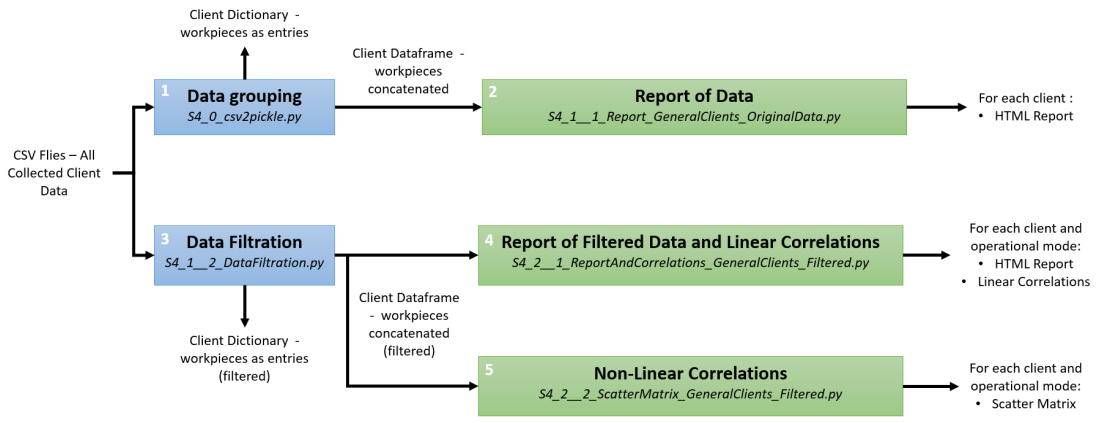


Figure 4.1: EDA Development - Simplified Block Diagram.

4.1 Evaluation and Filtration of data set's features

With the DataFrames created and saved, an exploratory analysis has been executed over the three main DataFrames that contain the information on the customer/machine level. The goal of this analysis is to get a general overview of the data for each customer and be able to see things that are common for all the different jobs or hardened pieces. By doing this first analysis, it is possible to detect unchanged/irrelevant values in the data set, that will lead to a filtration and complexity reduction of it. This first report-analysis is developed using the file named *S4_1___1_Report_GeneralClients_OriginalData.py*.

After the first Exploratory Data Analysis, it has been found that not all the variables collected are needed and that it would be interesting to have some additional calculated values. In this regard, different decisions have been taken according to the problems observed in each variable of the data set. The working Excel table created can be found in Appendix D, while in this section the most important conclusions are highlighted

There is a large group of variables that never change its value, not among the same client and sometimes not even among the three different clients. Therefore, these constant variables will not bring any relevant information in a later analysis, so they have been excluded from the data set. These variables or groups of variables and a short description of their behaviour are now listed:

- NC_Active: Always true, indicating that the machine always has one job/recipe selected.
- RobotInUse: Always false, indicating that there is no robot controlled by the same system to place and remove the pieces in the machine.
- LubricationActive: Always false, indicating that there has never been a lubrication process of the motors active while the machine was hardening a piece.
- WP_IO and WP_INO: Always false, indicating that the work-piece counter is not working in the machines and not able to detect if the piece hardened received the right amount of power or not.
- HeatingCycleComplete: Always false. This variable comes from the converter only if the Heating Sequence function is active on it. In these machines, the heating sequence is controlled by the PLC, and therefore this function is not being used on the converter.
- Converter_WaterFlow: Only in one customer's machine (*Client 1 - Sinac 100/160 SM*) the data regarding the water flow (both 1 and 2) in the converter is transferred to the PLC, while for the other two this information is not transferred and is always 0. Therefore, for the client analysis, these variables will be kept for the client that reads them and removed for the other two clients.

- Converter_WaterTemp: Always 0, indicating that there is no sensor reading the water temperature in the converter.
- Quenching Water: The quenching water flow and water temperature from both lines (1 and 2) is only read in one customer (*Client 1 - Sinac 100/160 SM*). In *Client 3 - Sinac 100 PM* only the first quenching line is present but not the second. While in *Client 2 - Sinac 150/70 SMH* this is not read in the PLC, due to that they have an external cooling station with its control system. Therefore, the quenching variables will be kept only for those clients and those lines where there is information about them.
- Axis errors and warnings: For all the clients, there hasn't been any file that has arisen an error or a warning in none of the axes. Except from *Client 1 - Sinac 100/160 SM*, where 86% of the samples contain a Synchronous Warning in the C axis. Therefore, only this variable related to axes errors and warnings is kept from all and for the three clients.

In addition to those variables that do not change at all in the whole data set, there are some variables that even if they change, have been found not relevant for the study or are highly biased. Those variables, together with a description of them, are the following:

- NewPiece: This variable was used in the data collection system to know when to split the data in a new CSV file, due to that the information from a new piece was received. In practice, seeing this flag to one also corresponds to a reset of the counter, therefore it is irrelevant to keep it.
- Automatic: For each client, less than 0.1% of the hardening processes are run in manual mode instead of automatic. This can be simply for some adjustments in the machine or some tests and not the actual hardening of a piece. Therefore, the files that are run in manual mode will be discarded, making the automatic variable constant and able to be discarded.
- MachineIsEmpty: For two of the three clients, less than 0.1% of the time the machine is run empty, therefore with no piece to harden inside. In this case, the same criteria as for the above variable applies.
- FrequencySelected: *Client 1 - Sinac 100/160 SM* is the only customer that uses this option to select between a high or low frequency from the converter. Only 2% of the samples are actually on low frequency. Due to the frequency value being shown as a process value, it is not very interesting to know which frequency is selected in the program, especially if only one customer is using it and this value is very biased. Then this variable is removed.
- Process variables limits: All the limits related to water temperatures never change in all of the clients, and the ones related to water flows do not change very often. In addition, variables in the process will never go out of the limits without a stop

of the process. Therefore, it is not interesting to use them in the analysis, but just use the process variables itself.

- X axis process variables: The X movement (which indicates left and right movement when looking at the machine from the front side) is very rare and for very special applications. None of the customers use this axis and therefore it makes no sense to take into consideration its values. For *Client 1 - Sinac 100/160 SM* and *Client 3 - Sinac 100 PM* there is no change in acceleration, position or velocity in this axis, while there are some variations in *Client 2 - Sinac150/70 SMH*. This can be due to some manual adjustments. The three process variables related to the X-axis will be removed from the three clients.

Finally, a variable that would be interesting to have is the accumulated heating cycle, and therefore this variable is calculated and added to the data set. To calculate this value, a column with this variable full of 0 is first added to the data set. While the power of the converter is on, the timestamp difference between the actual point and the previous one is calculated (always 200ms) and this is summed to the total value of that row. In cases where the power is on and off several times during the hardening of one piece, the accumulated heating time never resets to 0, but it keeps its last value when the power turns off and continues summing from there when this turns on again.

Analysing more in detail the contents of the data set, it is observed that when the PLC sends the signal to the converter to turn off the power, it takes one step delay to see that the current of the converter is turned to 0. In addition, the value used as set point (current or power) is immediately turned to 0 on the communication protocol from the converter to the PLC, but this is not true in reality. The current has a descending slope when the converter is turned off. This results in a continued calculation of the value that is not used as a set point (current or power), which is sent to the PLC. Therefore, sometimes it can appear that for 0 current value, a power value exists or vice-versa. To solve this problem, due that it is not possible to know the real value of the other variable that is not calculated, when the current or power is 0, the other variable is set to 0 too. In addition, the last used set point value is always available from the PLC side, but this value has no meaning if the converter is off. So this value will also be set to 0 if the current or the power is 0.

The last adjustment is that in the serial converters (the ones present in the machines from *Client 1 - Sinac 100/160 SM* and *Client 2 - Sinac150/70 SMH*) when the output power is off, the frequency value of the converter acquires a maximum defined value. This value is not relevant while the power is off, due that there is no actual frequency value while there is no current output. Therefore, the same condition explained before is applied. When the current or power is 0, the frequency is also 0.

The filtration over the general data set is done using the Python code found in *S4_1___2_DataFiltration.py*.

An overview of the filtered variables and their adjustments is presented in Table 4.1

Table 4.1: Filtered and Adjusted Variables.

Name	**Units**	**Comments**
Counter	-	
TimeStamp	-	
JobName	-	Name of NC program
HeatingCycleTime	s	Calculated variable - Accumulated Heating Time
C_AxisMode	-	
HeatingOn	-	Power is activated 1 step after HeatingOn is True
Converter_Setpoint	%	
Converter_Power	kW	Modified to 0 when Converter_AC_Current is OFF
Converter_DC_Voltage	V	
Converter_AC_Current	A	
Converter_Frequency	kHz	Modified to 0 when Converter_AC_Current is OFF
Converter_WaterFlow1	-	ONLY Present in *Client 1 - Sinac 100/160 SM*
Converter_WaterFlow2	-	ONLY Present in *Client 1 - Sinac 100/160 SM*
Y_Axis_Acceleration	mm/min^2	
Z_Axis_Acceleration	mm/min^2	
C_Axis_Acceleration	mm/min^2	
Y_Axis_Velocity	mm/min	
Y_Axis_Position	mm	
Z_Axis_Velocity	mm/min	
Z_Axis_Position	mm	
C_Axis_Velocity	$^\circ/min$	
C_Axis_Position	$^\circ$	
Inductor_WaterFlow	l/min	
Inductor_Temperature	$^\circ C$	
BusBar_WaterFlow	l/min	
BusBar_Temperature	$^\circ C$	
Quench1_WaterFlow	l/min	NOT Present in *Client 2 - Sinac150/70 SMH*
Quench1_Temperature	$^\circ C$	NOT Present in *Client 2 - Sinac150/70 SMH*
Quench2_WaterFlow	l/min	ONLY Present in *Client 1 - Sinac 100/160 SM*
Quench2_Temperature	$^\circ C$	ONLY Present in *Client 1 - Sinac 100/160 SM*
EnergyAlarm_Low	-	
EnergyAlarm_High	-	
Converter_Error	-	
C_Axis_Warning	-	

4.2 Correlation Analysis Development

After the filtration is done, a second Exploratory Data Analysis needs to be carried out on the filtered data set. This second analysis will have the scope to find out about the correlations between variables, both linear and non-linear ones. The two Python files used in

this section can be found in the following paths: *S4_2___1_ReportAndCorrelations_GeneralClients_Filtered.py* and *S4_2___2_ScatterMatrix_GeneralClients_Filtered.py*.

To understand better the non-linear correlations, these need to be plotted, to visually observe how two variables behave against each other. To achieve this, scatter plots for each pair of variables are needed. For this matter, a function called *pairplot* from the *seaborn* Python library has been used [36]. This tool provides a scatter plot matrix over a given data set, adjusting resolution and element positioning according to the size of the data set. This library gives a very clean and understandable output, which can be easily saved as an image for later analysis.

The data collected from the machines corresponds to a whole hardening process, which includes not only the heating itself but also the cooling of the hardened piece. Therefore, when analysing the whole data set, data from when the converter is ON and OFF is taken into consideration together. For a better understating of the data set, this has been divided into two parts: when the converter is ON and when the converter is OFF, which indicates if there is heating being applied or not. Then, a correlation analysis of the three data sets (including the general data set with both modes) is performed and its results are compared.

From the developed analysis, for each client and mode (ALL, ON, OFF) a correlation table with all the linear correlation coefficients is generated. It is known that the closer a correlation coefficient is to 1 (in absolute value), the stronger the correlation between the two variables is. Therefore, as used in many examples, such as in [37], a threshold of 0.6 will be used to determine strong correlations. All correlations which have a coefficient with an absolute value greater than 0.6 will be put together in another table. Then, the strong correlations of the three modes should be gathered together and compared between each other. This is because it can be that the same correlation exists in the three modes with a high value, or that the correlation is strong in some modes and weak in others.

To achieve the desired table for each client, the following steps using an Excel sheet with some functions have been executed:

1. The correlated variables, with a strong correlation, that are present in at least one of the three modes have been combined in a unique table (just the variables' names, not the correlation values). Due that some of the correlations are present in more than one mode, this table will have duplicates.

2. The data has been structured as a table and the duplicates were easily removed with an integrated Excel function. The remaining table indicates the strongly correlated variables that appear at least in one of the three modes.

3. An extra variable is created by the combination of the two correlated variables to have a unique identifier for each correlation. The variables are created as follows: $= Var1\&"|"\&Var2$.

4. The correlations tables of all the three modes have also been opened and the third variable described before was also generated in each one of them.
5. The correlation values for each pair of variables are obtained from the general correlation values through the *VLOOKUP* function. For this function, the unique identifier is used as a reference and the general correlation tables of each mode as a search table.
6. A table with the correlation values for each mode of each client is obtained. Each pair of correlation variables will have at least one mode with a strong correlation.
7. A formatting is applied on all the correlations, so that all the correlation values with an absolute value smaller than 0.6 are marked with a yellow background, to easily identify if a strong correlation is present in one or several modes.
8. The unique identifier column is now no longer needed and two extra columns to insert comments regarding the correlations are added.

Once this table is generated, it is possible to do a combined correlation analysis using also all the scatter plots collected in a scatter matrix. From this analysis, four different cases can be expected:

1. Non-linear correlation: This correlation is visible in the scatter matrix but not in the linear correlation value.
2. High linear correlation: This correlation is shown both in the scatter matrix and in the linear correlation value.
3. Localized linear-correlation: This correlation is only shown in the linear-correlation value but not in the scatter matrix. This is because, in the scatter matrix, all the data is shown (outliers included), therefore it can seem that the data is spread in the whole graph, but maybe its density is not the same in all the areas. Therefore, the linear correlation value will output a high correlation number, due to the majority of the data points being placed in a certain location and highly correlated, while in the scatter matrix, the plot is not big or detailed enough to show that correlations if points with very different values are present.
4. No correlation: None of the two systems will show any correlations.

4.3 Common Strong Linear Correlations

The tables containing all the strong linear correlations and comments about them, as well as all the generated scatter matrices, are found in Appendix E. This section will focus on the description and discussion of the most relevant or unexpected correlations.

The first thing is to have a look at the correlation variables that are present in all three customers, to see if it is possible to generalize the functioning of the different customer's machines. In Table 4.2 (Excel generated), a table with these correlations is shown. In this table, the correlations for each one of the three modes of each client are shown, marked with a yellow background the ones that are under 0.6 in one of the three modes. In the comments sections, a background colour code is also used. The ones that have a grey background, do not have any physical/relevant meaning. The green ones are relevant correlations but are known because of the nature of the machine. The orange ones are the more interesting ones because they may be dependent on the machine/converter type or the customer operation.

Table 4.2: Common strong correlations for all the three clients.[1]

Variables		Correlations									Comments	
		Client 1			Client 2			Client 3				
Variable 1	Variable 2	ALL	ON	OFF	ALL	ON	OFF	ALL	ON	OFF	Observations	Seen in Scatter Matrix
Converter_Setpoint	Converter_AC_Current	0.9988	0.9684	-	0.9369	0.8552	-	0.9192	-0.3074	-	- Setpoint and output current behave in the same way (serial). - Setpoint and AC Current turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information. (parallel)	Yes / Not so clear (parallel)
Converter_Setpoint	Converter_Power	0.9823	0.5386	-	0.7770	0.4886	-	0.9331	-0.0767	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the setpoint goes to 0 and so does the power. Not interesting	Yes / Not so clear (parallel)
Converter_Power	Converter_AC_Current	0.9817	0.5321	-	0.8306	0.6139	-	0.9953	0.9398	-	Power and current behave in the same way.	Not so clear / Strong linear correlation (parallel)
Converter_AC_Current	HeatingOn	0.9592	0.2610	-	0.9797	0.2699	-	0.9959	0.7588	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Converter_Setpoint	HeatingOn	0.9581	0.1459	-	0.9812	0.0402	-	0.9906	0.1083	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Converter_Power	Converter_Frequency	0.9515	-0.2348	-	0.2675	-0.7194	-	0.9326	0.1300	-	- Power and Frequency turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information. (Client 1 and 3) - Different Max. Output Power according to frequency Range selected in machine (higher frequency range ,smaller output power).(Client 2)	Not so clear / Strong logaritmic correlation (parallel)
Converter_Power	HeatingOn	0.9435	0.1713	-	0.9612	0.0707	-	0.9945	0.7461	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Counter	HeatingCycleTime	0.8669	0.9724	0.8550	0.9178	0.9989	0.4991	0.8664	0.9279	0.8533	The Heating Cycle Time variable it increases with the counter if the converter is ON. Artifical variable and no physical information between these two.	Yes
Converter_DC_Voltage	HeatingOn	0.8550	0.0332	0.0081	0.7299	0.0357	0.0073	0.9963	0.7710	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Inductor_WaterTemp	BusBar_WaterTemp	0.8525	0.8759	0.8261	0.7628	0.7988	0.7026	0.8149	0.8861	0.8846	The water temeprature in the Bus Bar and in the Coil behave in the same way. Same pump.	Yes
Counter	Z_Axis_Position	0.7308	0.5929	0.7455	0.6581	0.6950	0.4906	0.4729	0.7065	0.2855	The counter indicates the "timestamp" of the process, the axis position will increase while the counter goes up. No physical information.	Yes
HeatingCycleTime	Inductor_WaterTemp	0.6688	0.8442	0.6020	0.7978	0.8405	0.5714	0.1598	0.8167	-0.0723	The water temeprature in the coil increases over time while heating is ON	Yes
Counter	Inductor_WaterTemp	0.5293	0.7747	0.4479	0.6998	0.8356	0.0228	-0.1160	0.7313	-0.4009	The water temeprature in the coil increases over tthe process (specially when converter is ON)	Yes
Converter_DC_Voltage	Converter_AC_Current	-0.7540	-0.1203	-	-0.6645	-0.2629	-	0.9114	-0.1468	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the current goes to 0 and the voltage to a higher stable state (serial) or to OFF (parallel). Not interesting	Not so clear / Strong linear correlation also when ON (parallel)
Converter_Setpoint	Converter_DC_Voltage	-0.7543	-0.1210	-	-0.6498	-0.2274	-	0.9812	0.7208	-	- This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the setpoint goes to 0 and the voltage to a higher stable state. Not interesting (serial) - Normally setpoint is a current or power setpoint, but in the parallel converter is the voltage that is controlled. Therefore directed related with voltage (parallel)	No / Yes (parallel)
Converter_Power	Converter_DC_Voltage	-0.7600	-0.2484	-	-0.7469	-0.4500	-	0.9257	0.0376	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the power goes to 0 and the voltage to a higher stable state (serial) or to OFF (parallel). Not interesting	Yes/ Strong lienar correlation also when ON (parallel)

Starting to briefly comment on the green correlations, it is observable that the power delivered by the converter and the current in the coil are strongly related for all the clients, as is expected because of the working principle of induction heating. Regarding the cooling systems, it is understandable to see that the water that cools down the coil increases in temperature over time, especially when the converter is ON, due to the power

[1]The table is present in a bigger format in Appendix E.4.

dissipation of the process. In addition, the water temperature in the coil and in the bus bar also behaves in the same way, because it is the same water pump that is delivering water in serial to both systems.

Continuing with the correlations that do not have a lot of significance, it is noticeable that all the converter variables are related to the *HeatingOn* variable when analysing the whole data set (without splitting). This simply means that when the PLC gives the signal to turn on, to the converter, this does so (with a time delay), and vice-versa. Therefore, a change in this variable will be reflected in a change from 0 to an X value of power, current, set point and voltage. This is not relevant, because it does not represent a process or machine correlation, but simply that the communication protocol between the PLC and the converter is working properly.

Finally, it is relevant to discuss the variables which result in a strong correlation in the three clients, but do not behave in the same way in all of them. An example of this is the relationship between the set point and the current or the DC voltage of the converter. In serial converters, the control system controls the current in the coil to obtain the desired set point, while in parallel converters the voltage on the output capacitor is the controlled one, giving a relationship with the DC voltage value. That is the reason why for the first two clients, there is a high correlation between set point and current when analysing the ON mode of the converter, but in client 3 this correlation appears between set point and DC voltage.

Another interesting correlation to analyse is the relationship between the converter power and frequency. It is not expected a direct linear relationship between these two variables when the output of the converter is turned ON, but a negative linear correlation is found for Client 2. To understand this relationship it is needed to remember from Table 3.1 that this client is the only one that has an SMH converter, which is a serial converter that can operate in two very different frequency ranges. These converters have a maximum output power that depends on the selected frequency range, in this converter, the maximum output power in the lower frequency range (7-10 kHz) is 150 kW, while in the higher range (100-150 kHz) is limited to 70 kW. This means that if the machine runs maximum power at a higher frequency, this will be lower than the power when it is run at a lower frequency. This specification could be the explanation for the negative correlation found between these two variables only in this client.

4.4 Relevant Non-Linear Correlations

When analysing the scatter plot, it is noticeable that sometimes the linear correlations are not so easy to see, but some other correlations can be discovered. When analysing the scatter matrix of Client 3 while the power is ON, which operates with a parallel converter,

some strong correlations between converter variables are shown. These correlations are presented in Figure 4.2.

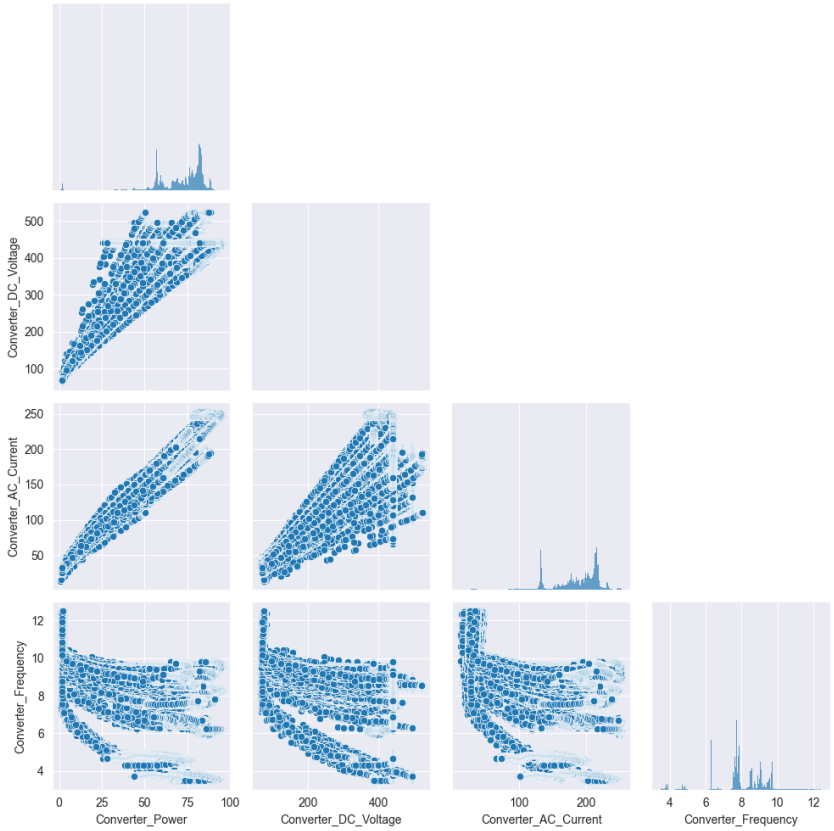


Figure 4.2: Relevant correlations from Scatter Matrix - Client 3.[2]

In the plots, three strong linear relationships are present between AC Current and Power, AC Current and DC Voltage and between DC Voltage and Power. It is remarkable that out of these three strong linear correlations, only the one between AC Current and Power is also shown in the linear correlation calculations. For the other two correlations, it is observable that multiple straight lines are present in the plot. This can be related to the fact that for different jobs or configurations, the machine follows a line with a different slope for the two variables under analysis. It is also observable that in some samples there is no correlation, but just a vertical or horizontal line (maybe due to some errors). This will imply that when calculating the linear-correlation value, this will not be relevant. Therefore, having the plots can help identify these correlations.

On the other hand, three strong logarithmic correlations are visible in the mentioned figure. These correlations correspond to the relationship between Frequency and Power,

[2]Histograms are shown in the diagonal of the matrix

and DC Voltage and AC Current. These correlations are due to the behaviour of the control system of the converter and how this finds the resonance frequency for the piece under treatment. The system always starts with a predetermined high frequency value which decreases over time until the resonance frequency is reached. The converter has some limitations regarding the power that can supply at frequencies which are not resonant. Therefore, until this one is reached, the three mentioned values will gradually increase as the frequency decreases and gets closer and closer to the resonant frequency of the workpiece. This behaviour gives this decreasing logarithmic shape in the three correlations.

4.5 Summary

Even if many of the correlations found in this analysis are known, some others are discovered in addition to some other relevant information. This analysis shows the diversity of the data collected, even if the three customers have a very similar machine doing the same type of work (induction hardening). This gives an overview of the complexity when it comes to data analysis and data-driven model development.

The results of this analysis are the starting point for building a ML model based on this data. When doing this, it is important to analyse the correlations of each client and to take into consideration those variables which are highly correlated with each other. These combinations of variables should not be used together when building a data-based model. This is because one behaviour (described by two variables highly correlated) will have double so much weight in the model than other variables that can be also important. Therefore, it is important to do a final filtration of the data set before implementing any modelling algorithm.

It is also to remark that the data set under study contains time series data, which is a topic discussed in Chapter 2.4 of the article 'Data Mining Methods - Application in Metallurgy' [7]. There it is stated that:"The time series analysis allows identifying of regularities between variable's values from different time periods. The main goal of time series analysis is the prediction of future events (future variable's values) based on known past events (past variable's values)." Considering that, the next step will be to develop an error-prediction model using a ML algorithm, to use it for predictive maintenance of the system.

5 LSTM Neuronal Network - Error Prediction

For this study and due to the structure of the data set, an LSTM neural network has been selected as a ML method to use to build a model for error prediction. The aim is to predict some type of failures in the machine before they happen, exploring afterwards the possibility of how this information can be useful to the in-place system to avoid this failure from happening. The data set used for this study contains time series data, where future behaviour can be influenced by past occurrences, making this type of network the ideal choice to analyse its behaviour.

As stated in Section 1.3, the article[6] explains that the methodology for data mining and analytics is divided into four main groups. For the development of this and the next chapter, similar blocks have been used to process the data and develop the LSTM model. In Figure 5.1, a simple block diagram of the main steps and outputs of this development process is presented. This block diagram does not include the Problem Selection chapter, which is not a direct part of the LSTM development.

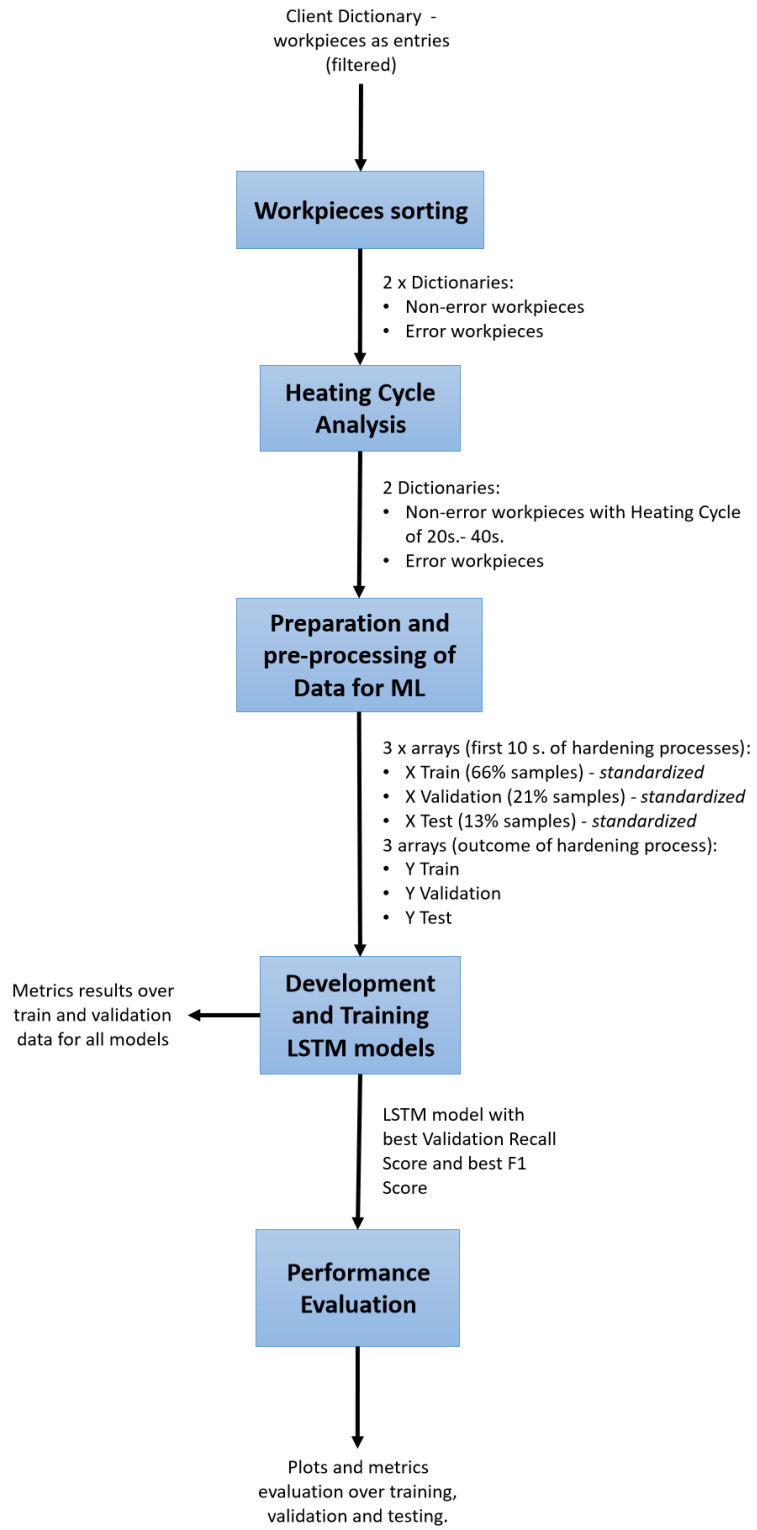


Figure 5.1: LSTM Case 1 Development - Simplified Block Diagram.

For the code development present in this and the next chapter, some articles and step guides have been used as references and inspiration. The website called "Machine Learning Mastery"[38] contains several relevant articles and examples about LSTM NNs, other Machine Learning models development, interpretation of results, metrics and parameters explanations, etc. From that website, the main article used is the one titled *"How to Develop LSTM Models for Time Series Forecasting"*[39]. In addition to that article, another one (from a different website) named *"LSTM for Predictive Maintenance on Pump Sensor Data"* [40] has also been of interest.

5.1 Problem Selection

All the figures present in this section, can also be found under *LSTM_Files/DataPreparation*.

As observed during the exploratory analysis of the data, each machine has its singularities which makes it very difficult to create a general model that can uniquely describe all the machines. In addition, within each machine of each client, many parameters are highly influenced by which job/configuration is selected in the machine. Therefore, the first step will be to analyse single jobs within each client and see if it is possible to build a data-driven model for that specific case.

The study selected is to predict failures in the machine or the process before they occur. To identify these errors correctly, a data set which contains enough error samples is important. Therefore, the first thing to analyse is which job within each client is the one that will provide a more balanced/varied error data set. A table for each client will be created indicating how many pieces were hardened using that configuration, and how many of those had one of the following relevant errors: converter errors, energy alarms (low or high) and axis errors/warnings. This is achieved using the following Python file: *S5_1___1_NumberOfErrors.py*.

A summary table for each client is created to have an overview of the errors present and to see the available possibilities. These tables are now presented in Tables tables 5.1 to 5.3 (Python generated).

Table 5.1: Error Type Counting - Client 1.

Error Counting Client1

JobName	Files	Converter_Error	EnergyAlarm_Low	EnergyAlarm_High	C_Axis_Warning
Job1	575	0	575	0	574
Job2	1078	2	1077	0	0
Job3	1896	0	1896	0	0
Job4	928	0	928	0	926
Job5	2962	6	2962	0	2941
Job6	412	5	412	0	400
Job7	323	0	323	0	318
Job8	205	1	5	0	191
Job9	304	1	304	0	301
Job10	101	1	101	0	98
Job11	9321	1	8039	0	9282
Job12	108	0	108	0	104
Job13	50	0	50	0	43
Job14	204	0	204	0	198
Job15	47	0	47	0	41
Job16	64	2	1	0	0
Job17	534	1	534	0	527
Job18	326	1	326	0	319
Job19	804	0	804	0	798
Job20	510	0	510	0	505

Table 5.2: Error Type Counting - Client 2.

Error Counting Client2

JobName	Files	Converter_Error	EnergyAlarm_Low	EnergyAlarm_High	C_Axis_Warning
Job1	498	0	0	0	0
Job2	41	1	0	0	0
Job3	102	0	0	0	0
Job4	1748	30	0	0	0
Job5	1241	0	0	0	0
Job6	3175	21	81	0	0
Job7	37	1	0	0	0
Job8	286	2	0	0	0
Job9	111	2	0	0	0
Job10	2033	14	0	0	0
Job11	248	0	228	1	0
Job12	1027	6	610	0	0
Job13	24	0	24	3	0
Job14	107	0	100	4	0
Job15	30	0	30	0	0
Job16	6	4	5	0	0
Job17	5237	10	4025	0	0
Job18	892	6	191	0	0
Job19	966	2	814	1	0
Job20	72	1	69	0	0
Job21	82	3	0	0	0
Job22	47	4	0	0	0
Job23	203	0	0	0	0
Job24	40	1	0	0	0
Job25	92	0	0	0	0
Job26	323	4	0	0	0
Job27	72	0	0	0	0
Job28	17	2	0	0	0
Job29	109	0	0	0	0
Job30	51	0	0	0	0
Job31	116	1	0	0	0
Job32	65	0	0	0	0
Job33	69	0	0	0	0

Error Counting Client3

JobName	Files	Converter_Error	EnergyAlarm_Low	EnergyAlarm_High	C_Axis_Warning
Job1	569	0	0	0	0
Job2	1	0	0	0	0
Job3	335	1	0	0	0
Job4	294	0	0	0	0
Job5	95	0	0	0	0
Job6	1	0	0	0	0
Job7	82	0	0	0	0
Job8	88	0	0	0	0
Job9	102	0	0	0	0
Job10	52	0	0	0	0
Job11	7	0	0	0	0
Job12	6	0	0	0	0
Job13	18	0	0	0	0
Job14	58	0	0	0	0
Job15	203	0	0	0	0
Job16	47	0	0	0	0
Job17	13	0	0	0	0
Job18	88	0	0	0	0
Job19	328	1	0	0	0
Job20	1190	0	0	0	0
Job21	17	0	0	0	0
Job22	1	0	0	0	0
Job23	30	0	0	0	0
Job24	40	0	0	0	0
Job25	36	0	0	0	0
Job26	225	0	0	0	0
Job27	60	0	0	0	0
Job28	36	0	0	0	0
Job29	107	0	0	0	0
Job30	31	0	0	0	0
Job31	3	0	0	0	0
Job32	15	0	0	0	0
Job33	12	0	0	0	0
Job34	10	0	0	0	0
Job35	21	1	0	0	0
Job36	16	0	0	0	0
Job37	206	0	0	0	0
Job38	155	0	0	0	0
Job39	101	0	0	0	0
Job40	84	0	0	0	0
Job41	406	0	0	0	0
Job42	153	0	0	0	0
Job43	108	0	0	0	0
Job44	199	0	0	0	0
Job45	204	0	0	0	0
Job46	197	0	0	0	0
Job47	61	0	0	0	0
Job48	12	0	0	0	0

Table 5.3: Error Type Counting - Client 3.

As shown before in the EDA section, only one type of axis warning is present in one of the clients and is the one corresponding to the C axis (rotational axis). Due to a large number of files containing this warning, it indicates that it is not a working/functional problem, but a warning that does not prevent the machine from running in normal conditions. In addition, it is only in one client and therefore not so interesting to study.

When looking at the energy alarm errors, it is observable that for almost all jobs in Client 1 and some of the jobs in Client 2, almost all the workpieces present this error. To understand this error better, an explanation of how this error is generated is given. When the power from the converter is on and the workpiece is being heated (or hardened, in this case), a function calculates the amount of energy delivered to the workpiece over the heating time. When the heating cycle is completed, the energy delivered to this piece

is compared with a user-preset value for the specific workpiece. This value is specific for each type of workpiece under treatment and must be configured for each of the different jobs. If the energy delivered to the workpiece is lower or higher than the one expected (within an acceptable range), this alarm arises at the end of the process. In this case, the mentioned clients are not setting this parameter correctly for some of the jobs or using a standard/the same parameter for all of them. This will then always create this alarm, which is not real. This could be because the clients have their own method of analysing if the hardening was correctly done or not for some of the workpieces, and they are not using this machine function. Therefore, this error cannot be used as a target to predict, because first it would be need to sort these errors between: "real/not real" energy alarms.

Finally, the last type of error present in the clients is the converter error. These errors, also called alarms, are the ones coming from the frequency converter and if one of them arises, the output power will automatically stop, meaning a stop in the hardening process. These errors are therefore more interesting to study because they are critical. Analysing the error distribution in the different clients, it is observable that the job that contained more workpieces with an error is Job 4 from Client 2. Therefore, this job will be the one selected for this study.

The converter error variable is a 16-bit (2 bytes) variable, where each one of the bits encodes a possible error. This means that if the corresponding bit is 1, the error associated with that bit is raised in the converter. Knowing the structure of the message sent from the converter to the PLC it is possible to create a decoding function that outputs the error name/s for each converter error variable received (*S5_1___2_ErrorWarningDecoding.py*). For the selected job, it is now interesting to see how is the error type distribution of the 30 workpieces that have an error. Even if it is the first error the one that stops the converter, two errors may originate at almost the same time, ending up with a workpiece that contains more than one error. Therefore it is possible that the counting of type errors is greater than the number of workpieces that have an error, as it is in this case, shown in Figure 5.2. The mentioned figure is obtained using *S5_1___3_TypeOfCOnverterErrors.py*

Error Types Client2-Job4

Error Name	Amount
Water Flow Low 1	12
Water Flow Low 4	2
DC Voltage Too Low	8
External Fault	8
Inverter Common	2
Overcurrent Driver	2

Figure 5.2: Error type distribution for the selected client and job.

This happens because the Overcurrent Driver is in this case always coming together with Inverter Common (2 workpieces). These two errors are related to the driver card that controls the transistor switching. The Overcurrent Driver alarm comes directly to the control system of the converter, and if other alarms are generated, then the Invert Common alarm is generated. This alarm means that if the operator wants to know more details about all the alarms generated in the driver card, it is needed to check the LED indicators in this card. In addition, for the other 2 workpieces, the Waterflow Low 1 error comes together with the External Fault Error. This can be due to a fault coming from the external cooling unit or a water pump that supplies water to the converter, which is read in the converter as an External Fault, but at the same time, it causes a water flow below the allowed limit. This explains that 4 workpieces have 2 errors instead of one, explaining the distribution of 34 errors over 30 workpieces with an error.

To do an error prediction, it is important to decide which of the errors the algorithm is going to predict. It is important to have the largest number possible of error examples, to train the NN properly. Analysing the distribution, it is observable that the most common error is Waterflow Low 1. As explained, the water that cools down the converter comes from an external pump and this error is directly related to it. Therefore, it would be irrelevant to analyse process variables (that is what is available) to predict this error. The same occurs with the External Fault error. Therefore, the next larger error is the DC Voltage Too Low, which can be related to the process itself and the main power supply of the converter.

Therefore, it is decided to build an LSTM NN that will try to predict a DC Voltage Too Low error in the machine before this occurs.

5.2 Data Selection

The first step is to decide which variables from the data set will be used to build the desired model. Therefore, using the EDA method, a report generated by *ydata-profiling* [35] will be created for that specific job. The report is originated using *S5_2___1_ReportData_SpecificJob.py* and it can be found in *LSTM_Files/DataPreparation/Report_Client2_Job4.html*.

From this report, it is observable that there are some constant variables, such as: JobName, C_AxisMode, Y_Axis_Velocity, EnergyAlarm_Low, EnergyAlarm_High and C_Axis_Warning. These variables will not be taken into account for the model development. In addition to those that are constant, some variables are highly related to each other and therefore it is better to remove them. From these pair of variables, just one of them is kept and the other one is removed, except for the converter variables, which are all kept. This is because, even if they are numerically related, they give important

information about how the frequency converter is working and can be relevant when analysing errors occurring in it. To reduce even more the number of variables used in the model development, for each axis, only one variable between acceleration, velocity and position is kept, which in this case is position. Finally, it is to mention that the Converter Setpoint is a variable that is set by the operator and not a feedback process variable, therefore it will also not be used for the model development.

This means that the variables that will be kept for the LSTM NN development will be the 11 following:

- Counter.
- Converter_Power.
- Converter_DC_Voltage.
- Converter_AC_Current.
- Converter_Frequency.
- Y_Axis_Position.
- Z_Axis_Position.
- C_Axis_Position.
- Inductor_Waterflow.
- Inductor_WaterTemp.
- Converter_Error.

The development that follows from here until the end of the chapter, is present in the Jupyter notebook named *S5_2___S5_3___S5_4___S5_5_LSTM_Case1.ipynb*.

Now that the variables are selected, from the general dictionary containing all the jobs related to Client 2, the workpieces related to Job4 that do not have an error or that contain the error under study (DC Voltage Too Low) will be extracted and separated into two different dictionaries (with error and without error).

After that is done, the heating cycles of each one of the workpieces will be summarized using a histogram, grouping them by workpieces that have a heating cycle time within 20 seconds of each other. The expectation is that almost all the workpieces should be in the same group because the understating of the job/recipe is that the same configuration is applied and the same or very similar workpiece is hardened. Analysing the obtained graph it is immediately observable that this client is not using this job configuration as expected, due that the heating cycle times are spread from 1 s. until around 360 s. This indicates that the same configuration with a manual adjustment of some parameters is

used for different purposes, making it even more difficult to have a general model or understanding of the data.

For analysing if this always happens, these histograms are plotted using *S5_2___2_HeatingCyclesAnalysis.py* for all the different jobs of each client, placing the results in *LSTM_Files/DataPreparation/HeatingCycleTimes*. Having a fast overview of the histograms for each client, it is seen that for almost all the configurations used in the three clients, ENRX's understating of job/recipe matches the reality. One job has all of its work pieces grouped in some histogram bars. It is not expected that all the hardening processes will have the same heating cycle over 3 months of production, due to some quality adjustments or changes of requirements in the hardening result being needed. But it cannot also be a large spread of values, as it happens for example in Job 4 and 6 of Client 2, due to this will remove importance to the configuration selected. Therefore, this client may have used the two mentioned jobs to test different workpieces.

Even if there is no guarantee that the analysis will be on the same workpiece, to have a data set that is expected to be more uniform, workpieces with no errors that have a heating cycle time between 20 and 40 seconds will be selected. This is because there is quite a large number of files within this range and it is a long enough heating cycle time to use a significant part of it as a prediction data set using an LSTM NN. It is known that the hardening result from two heating cycles with 20 seconds of difference will for sure not be the same, but due to that this job was the one containing the larger number of errors (which is the aim of the model), this selection of data is done. Regarding the pieces which contain some errors, the heating cycle times will not be considered and all the samples will be taken into account, due that the number of them is already small. It is important to remember that the error under analysis is DC Voltage Too Low, meaning that the heating cycle time of the workpiece itself should not play a big role in identifying it.

By doing this, the result is two dictionaries containing the following:

1. Non-error files: 348 with heating cycle times between 20 s. and 40 s.
2. Error files: 8 with heating cycle times between 10 s. and 130 s.

5.3 Data Preparation

To structure the data for the training and development of an LSTM NN, the data set will be split into three main groups: train, validation and test. The train and validation data set will be used to build and validate the model, while the test data set will be used to evaluate the model's performance. Due to that, the data is highly unbalanced, having a much larger number of workpieces with no errors than with errors, the division of the data set will be done in a semi-manual mode. Around 70% of the non-error data will

be used for training 20% for validation and the rest 10% for testing. While for the error data, approximately 60% of the data will be used for training, 20% for validation, and 20% for testing.

First, for each one of the two dictionaries (files with errors and files with no errors) a random list is generated, indicating which keys of the original dictionary will be used for the train data set, which of the remaining are used for the validation data set and leaving the remaining ones for the test data set. This will give 6 dictionaries containing the following information:

- A training dictionary containing 243 non-error files.
- A training dictionary containing 4 error files.
- A validation dictionary containing 70 non-error files.
- A validation dictionary containing 2 error files.
- A testing dictionary containing 35 non-error files.
- A testing dictionary containing 2 error files.

Each one of the files contained in each one of the dictionaries corresponds to a whole hardening process of one piece, containing the 11 variables mentioned in Section 5.2 and several time samples. This is not the desired structure for an LSTM NN, therefore this needs to be adapted to the problem's needs. For this case, the first 10 seconds after the converter is turned on will be used to determine if the workpiece under heating will fail or not in the process due to a DC Voltage Too Low alarm. Therefore, for each pair of the dictionaries mentioned above, a 3D array will be created. Due to the data being highly unbalanced, to compensate for this gap between non-error and error files, the error data will be duplicated 10 times, ending up with 40 samples for training, 20 for validation and 20 for testing. These three arrays will be the X training, validation and testing arrays, which are the input arrays to the LSTM model and have the following structure:

- First dimension: Number of workpieces in the array (283 training, 90 validation, 55 testing).
- Second dimension: Number of samples of each workpiece/hardening process (50).
- Third dimension: Number of variables present in each workpiece/hardening process (11).

In addition to the X array, due that this is a supervised ML problem, meaning that the data has a label indicating the expected output, three one-dimensional Y arrays will be created (again, one for training, one for validation and one for testing). These arrays will have the same shape as the first dimension of the corresponding X array and will contain only boolean values. A true (1) value will mean that the workpiece has an error, and a false (0) will indicate that it does not contain any error.

In this way, the data set created will have the information of the first 10 seconds of each workpiece related to the outcome of the hardening process: DC Voltage too Low alarm or No Error. This will then have the right structure for an error prediction model because it will be able to determine a future error by only looking at the first samples of the heating process. Due to the arrays being manually created, all the non-error files will be located at the beginning of the array and the error files at the end. To make this distribution more uniform, the X and Y data sets will be randomly shuffled along the first dimension while keeping the correlation between each file and its labelled output. The samples inside each workpiece/file will not be shuffled as well as the position of each variable, but just the position of the file location/index within the array.

The last step before starting to build and train the LSTM model, will be to evaluate if the input data needs to be scaled or standardized, or if the output data needs to be encoded. In this case, the output is just a boolean value, meaning that the model output will be binary (1 or 0), making it unnecessary to encode it (which is usually done with classification problems that have more than 2 possible outputs). Regarding the input data, standardization has been selected. This is needed due that each one of the variables is measured in different units and therefore will have different ranges of values. If standardization is not implemented, a variable that has a higher value (just because of its nature) will acquire more importance than a variable with a smaller value. For each variable, standardization will centre and scale the data by subtracting each single value from the average and dividing the result by the standard deviation. By doing this, all the resulting values will have the same scale due to their mean value being 0 and their standard deviation being 1. This method is preferred against normalization, where the data is scaled between 0 and 1 because it is less sensitive to possible outliers in the variables. The general equation as well as the general data distribution after standardization is shown in Figure 5.3.

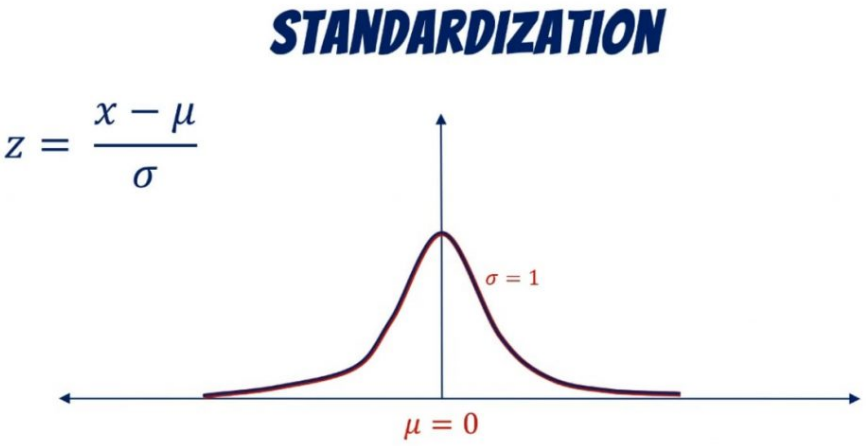


Figure 5.3: Standardization of Data - equation and distribution [41].

The best way to standardize the data is to fit the training data to the standardizer (to find the mean and standard deviation of it) and use those values to transform the other vectors. The results presented in this report are although executed by applying a fit and transform independently to each one of the vectors, this can be possible since the data distribution is very similar in the three vectors. In any case, an extra branch in the GitLab Repository named *stdr_fixed* can be found with this small detail fixed. The results obtained with this small fixed have not been included in this report, since they present almost the same information.

The data is now structured and pre-processed in the way needed. Training, validation and testing arrays with standardized data are randomly shuffled and ready to be used to design, train, validate and evaluate an LSTM NN model. These arrays have been saved and can be found in *LSTM_Files/LSTMVectors/Case1*.

5.4 Model Development

For this project, the LSTM model has been manually built using the *Sequential* model structure offered by *Keras* [42]. This structure allows the user to build its model, by deciding several configuration parameters. For this purpose, a building function has been developed to return an LSTM model based on the user input, which can decide the number of layers of the model and the number of neurons and activation functions in each one of the layers. In addition, the user also needs to specify the input and output shape of the model. With these parameters, the function creates an LSTM model with the selected configuration of LSTM layers and adds a standard output **Dense** layer with *Sigmoid* as the activation function for this final output layer of the model. According to the data values (which have been standardized) and because the output of the model should be either 0 or 1, this function is usually the one that will give better performance (in all types of binary classification problems). As shown in Figure 5.4 this activation function outputs a number between 0 and 1, having an input domain from - infinity to + infinity and high sensitivity between -2.5 and 2.5. In addition, all the internal LSTM layers (except the last one before the output layer) have a parameter called *return_sequences* set to True. This means that each memory cell will return all the hidden states of each time step, which is very helpful when stacking several LSTM layers and using them for sequence prediction, which is the case.

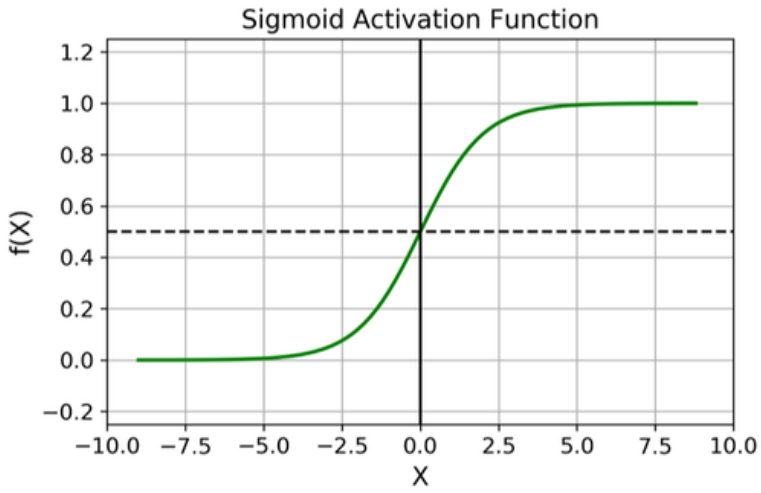


Figure 5.4: *Sigmoid* activation function [43].

Internally, the model loss will be calculated using the mean square error parameter, this parameter is obtained by calculating the average squared difference between the predicted values and the target values. Therefore, the aim is to have this parameter as close to 0 as possible as the training evolves, meaning that the predicted and target values are the same. Regarding the model performance, two parameters have been calculated which are: recall and precision. These two parameters are usually good metrics when handling classification problems, especially if they are focused on binary error prediction. The metrics aim to optimize the True Positives predictions, meaning as positive the presence of an error. Explained in a very simple way, precision is the metric that indicates how many errors predicted by the model were real errors, while recall indicates how many of all the real errors were correctly predicted. In addition, there is another interesting variable in classification problems which is called F1 Score. This variable is the harmonic mean of the precision and recall value. This value is not calculated over the training of the model, due that if in one of the training epochs, the model has a value of 0 in Recall and Precision, the F1 Score will not be a number. To understand better how these three metrics work, a confusion matrix with only the relevant metrics' formulas is presented in Figure 5.5

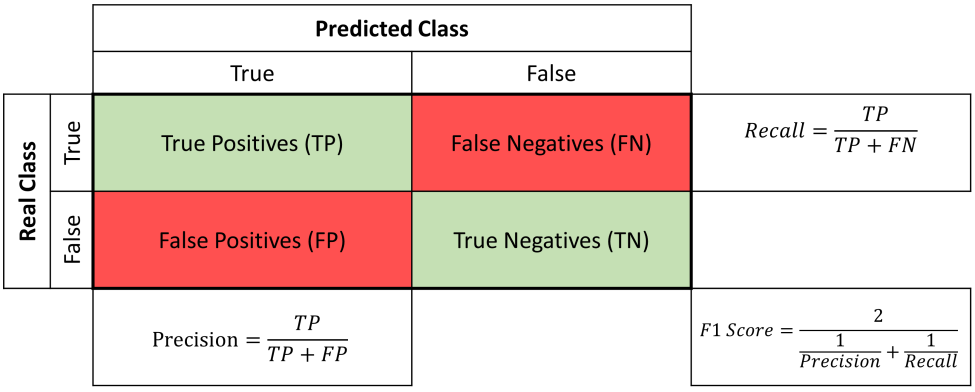


Figure 5.5: Confusion Matrix and relevant Performance Metrics.

To find the best model for the problem under study, 12 different sequential models will be built by trying different numbers of hidden layers, with different neuron configurations and using different activation functions in the hidden LSTM layers (for one model, the same function will be used in all the hidden layers). When having NNs with several layers, *ReLU* is usually the selected activation function, because it overcomes the vanishing gradient problem which usually appears with other activation functions, giving a very good performance with non-linearities. This is because, as shown in Figure 5.6 this function returns 0 for input values smaller than 1 and the same input value for positive input. This means that the partial derivative of the loss function will be either 1 (for positive values) or 0 for zero or negative values, preventing the gradient from vanishing [44], [45]. Just for comparison, *Sigmoid* will also be tested as a possible activation function in the hidden layers.

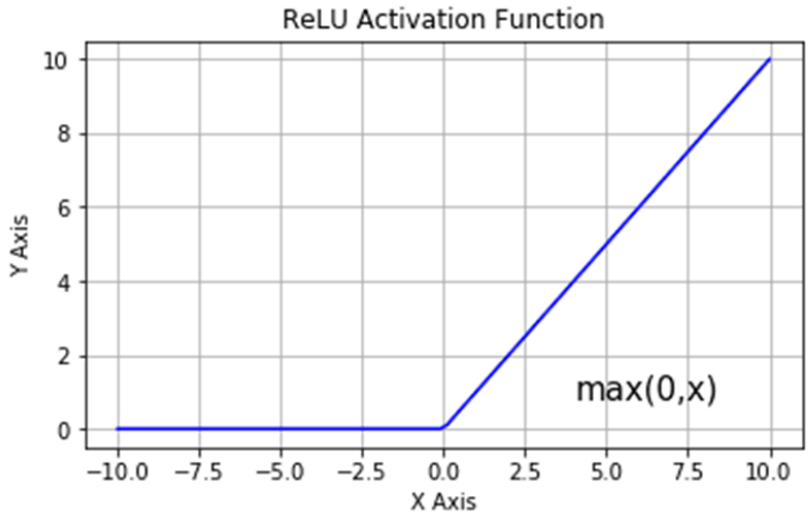


Figure 5.6: *ReLU* activation function [43].

Only 2 or 3 layers will be evaluated due that a larger number of layers in a problem that does not have so many variables and that it is not that complex, could lead eventually to an over-fitting. In [40] it is stated that: "The rule of thumb here is fewer hidden units (memory cells) than input features". Therefore, in this study, the number of neurons in each layer will never be greater than the number of input variables. In the example from the reference, the same number of units were used in each layer, but in this case, this will gradually decrease in each layer towards the number of output variables. For this problem, 11 input variables and 1 output variable are present. Therefore, three different neuron combinations will be tested for a 2- and 3-layer configuration.

- 2 layers: [10,3], [10,5], [10,7].
- 3 layers: [10,5,2], [10,7,5] ,[10,8,6].

Each one of these 12 models will be trained with a different number of epochs and batch sizes. The epochs are the number of times that all the data is passed through the model to improve it, and the batch size is the amount of data that is passed at the same time within each epoch. This means that if a model is trained for 50 epochs with a batch size of 20, in each epoch the data will be split into groups of 20 and the model will be gradually trained for each one of the batches 50 times. In this instance, each model has been trained across epochs ranging from 50 to 200, with an interval of 50 epochs between each, and using batch sizes between 20 and 100, with an interval of 20.

Different methods exist to execute this training, such as using the pre-build function *GridSearchCV* from *scikit-learn* [46]. In this function, the selected model is trained using a K-fold cross-validation method, using different training parameters' values, such as epochs or batch sizes, and returns the estimator which gives a better performance according to the selected scoring method.

This library has been tested and it can be very useful, but when having several model structures that need to be evaluated over several parameters and for different score performances, it is better to implement a semi-manual training method. This will allow a better understanding and manageability of the results and variables of interest. Therefore, 5 nested for-loops are implemented to evaluate each one of the possible combinations described above. For each one of them, the model will be compiled and fitted over the training data (using the validation data as a performance measure) for the selected combination of epochs and batch size. After each training, the validation recall and precision values of the last epoch will be stored and the validation F1 Score for the last epoch will be calculated. The last recall and last F1 score parameters will be the ones used to select the best 2 models of all. A higher F1 Score is usually the desired one, but this can sometimes mean a very good precision and a not-so-good recall. As a general rule, it is always better to have predicted an error when this was not going to happen than not predicting a real error that actually will happen (meaning a better recall). That is the

reason why the best recall model is also saved. The best out of these two models will be selected when evaluated over the test data.

The models are evaluated from smaller to higher complexity and size of training parameters. In this way, if two models have the same performance, the less complex model will be the one saved as the best model. The results obtained from this simulation are now shown in Table 5.4.

Table 5.4: Best Models Results - Case 1.

	Best F1 Score Model	**Best Recall Score Model**
Number of hidden layers	3	2
Number of neurons per layers	[10,5,2]	[10, 3]
Activation function in hidden layers	*ReLU*	*ReLU*
Output layer	1 Neuron - *Sigmoid*	1 Neuron - *Sigmoid*
Number of epochs	150	150
Batch size	20	60
Last Validation F1 Score	0.85	0.65
Last Validation Recall Score	1	1

The development and testing of this model can be executed by the reader using the file named *S5_4___S5_5_LSTM_Case1_ONLY_TrainingAndTesting.ipynb*, which takes as input parameter the previously saved vectors.

5.5 Model Performance Evaluation

All the figures present in this section, together with the dictionaries containing the best models presented in the above table, can be found under *LSTM_Files/Results/Case1*.

Analysing with details the results presented in Table 5.4, it is observable that both models do present a validation recall score of 1. The main difference is that the F1 score is better in one model than the other, meaning therefore that the precision of one of the models is better than the other. Even if the model that presents a higher F1 Score is slightly more complicated, due that it has an extra hidden layer in its structure, this will be the one selected to evaluate the test data. To analyse the performance of the selected model

along the epochs that it has been trained, plots showing the evolution of the loss and performance metrics are shown in Figure 5.7.

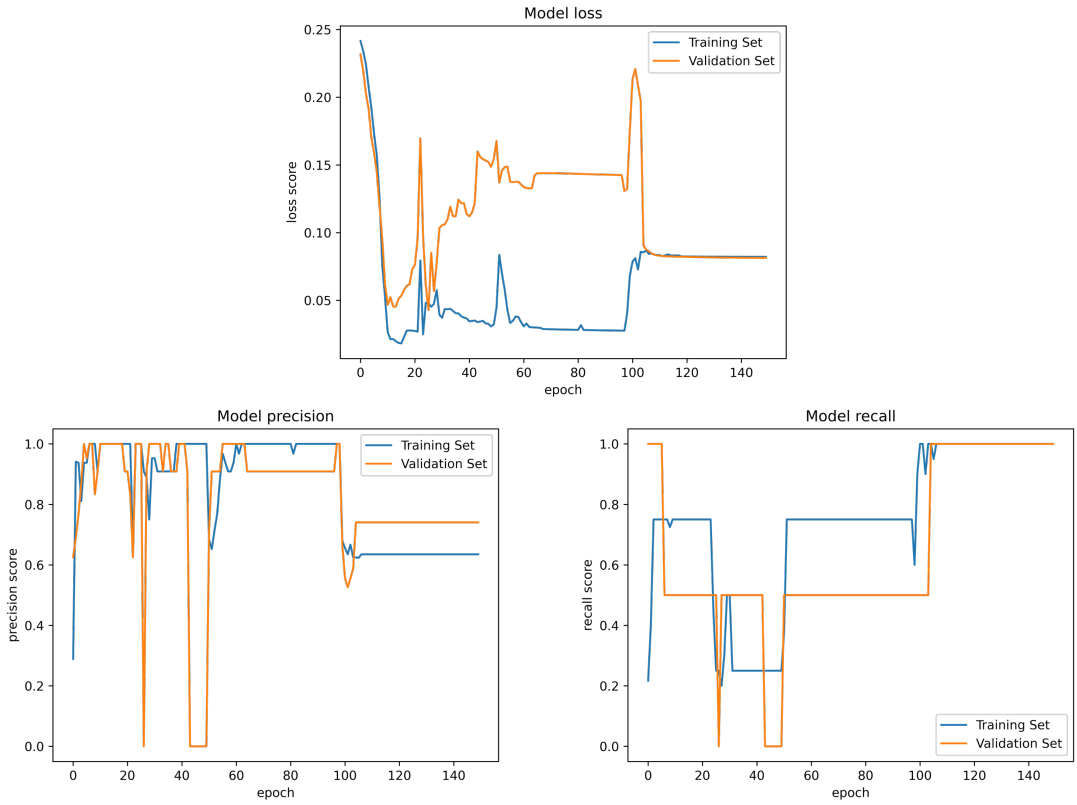


Figure 5.7: Performance Metrics of Best Model - Study Case 1.

Starting by analysing the model loss, the training and validation loss present an overall similar behaviour with some main deviations. For the first 10 epochs, the training and validation loss are decreasing together. After that, the validation loss starts to grow, having a big peak around epoch 20. This could be because when the model gets more complicated over the epochs, to optimize its behaviour with the whole data, errors can be introduced that will imply a sudden drastic change in the relevant metrics. After that peak, the loss of the two data sets keeps behaving in a very similar way, but with an offset of around 0.12 between each other. This means that the model is evolving in the right direction, but it is still not good enough to work on unknown data (validation). Finally, around epoch 100, a big change is introduced in the model, which slightly increases the loss value of the training data set, but drastically reduces the validation loss, making it converge with the training loss score. This model ends up with a loss score that is lower than 0.1, which is a very good and low enough value taking into consideration the complexity of the data and a highly unbalanced data set.

Having a look at the precision score of the model along the epochs, it is observable that both for the training and the validation set, it also has a similar behaviour, but with bigger changes in the validation data set. It is to remember, that even if the error data has been duplicated 10 times, to have 20 error samples in the validation data set, these are just 2 different samples. Therefore, the number of True Positives can drastically change in one epoch. If none of the two error types are detected correctly, independently of how many false positives (non-error files are classified as errors), the precision score will go to 0. This can explain the behaviour of the model going to 0 precision score over the validation data set in some of the epochs. As shown in the previous graph, the change implemented in the model around epoch 100, does decrease the training precision score, but it does stabilise this score for the validation data set around 0.75, being even higher than in the training data set, which is around 0.65. The fact that the validation score is higher, could be because the errors and data structure present in this data set are easier to interpret by the model than the 4 error types present in the training data set for the final model. A validation precision of 0.75 is not optimal but is an acceptable value for the case under study

Finally, looking at the recall score, which is one of the most interesting ones for this type of study, this starts with a very high value for the validation data set and a very low one for the training data set. In this metric, which only takes into account the error files, corrected or wrong classified, the fact that the data has been duplicated is even more visible. Even if there are 40 error files in training and 20 in validation, actually there are 4 types in training and 2 in validation. This means that this score can only take values of 0, 0.25, 0.5, 0.75 and 1 for the training set, and of 0, 0.5 and 1 for the validation data set. This is observable in the graph, and that is why big jumps are present along the different training epochs. Around epoch 100, the model seems to be able for the first time to get the recall score of both the training and validation data set to 1. This is an optimal result for the study case, but due to the limited number of error files present, this performance is not expected when analysing a whole new data set.

This model will be now tested against the testing data and the value of the three metrics mentioned above will be evaluated one more time. It is to remember that the test data set has never been seen by the model, simulating real-world data. When evaluating the model over a new data set, the output, due to the sigmoid function present in the output layer, can be any value between 0 and 1. Therefore, an optimal threshold needs to be found over the testing data to obtain a higher number of correct predictions. In this case, this threshold is found to be 0.45. The results of the predictions given by the model over the test data can be seen in Figure 5.8.

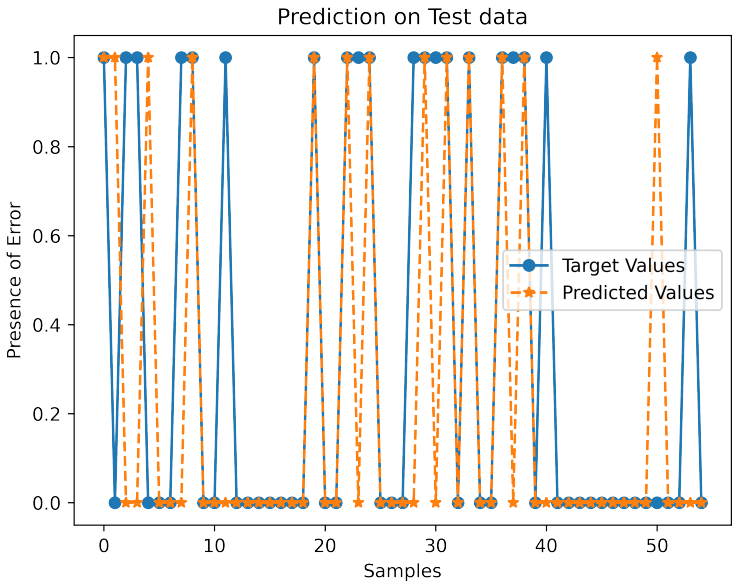


Figure 5.8: Model Predictions on test data - Study Case 1.

The model performance seems to be quite good, being able to identify correctly 42 samples of a total of 55, which corresponds to 76% of all the samples. The performance metrics over the testing data are presented in Table 5.5.

Table 5.5: Performance metrics over test data.

F1 Score	0.61
Precision	0.77
Recall	0.5

It is observed that the recall score over the test data has decreased to 0.5. One more time, the test data has 20 error samples, which only correspond to two different errors, meaning that the model is only able to detect one of them (10 samples). On the other hand, the precision score is quite good, being near 0.8, improving the overall F1 Score. Meaning that at least the model does not classify many non-error files as error ones. The model is not optimal over real-world data, especially if the focus should be on detecting all possible errors (optimizing the recall score), but the insufficiency of error data should be taken into account when analysing these results.

6 LSTM Neuronal Network - Improved Error Prediction

For the previous model to have high relevance in the in-place system, the control system of ENRX's machine should take into consideration the prediction made by the LSTM model in real time to try to compensate for that voltage loss. Unfortunately, that is not always possible, since the voltage loss is sometimes due to other machines in the production line. Furthermore, even if the model can predict the error before it happens, and it could stop the machine before that, the piece that was under hardening needs to be discarded. Sometimes, these pieces are very big and expensive, so the ideal case would be to not stop a hardening process in the middle because of an error, as that would mean a big loss for the customer.

Therefore, the study case now presented will try to predict the same error before the machine is turned on and the hardening process starts. This means that, instead of analysing the data from the beginning of the actual process, to determine if it will fail in the future, previous heating cycles will be analysed, to determine if the next one is going to fail. In this case, the hardening process will not start at all and the piece that needs to be hardened does not need to be discarded, giving the operator the possibility to monitor the power usage of other machines before starting the process again.

One more time, a simplified block diagram containing the main steps and outputs in the development process of this LSTM algorithm is presented in Figure 6.1.

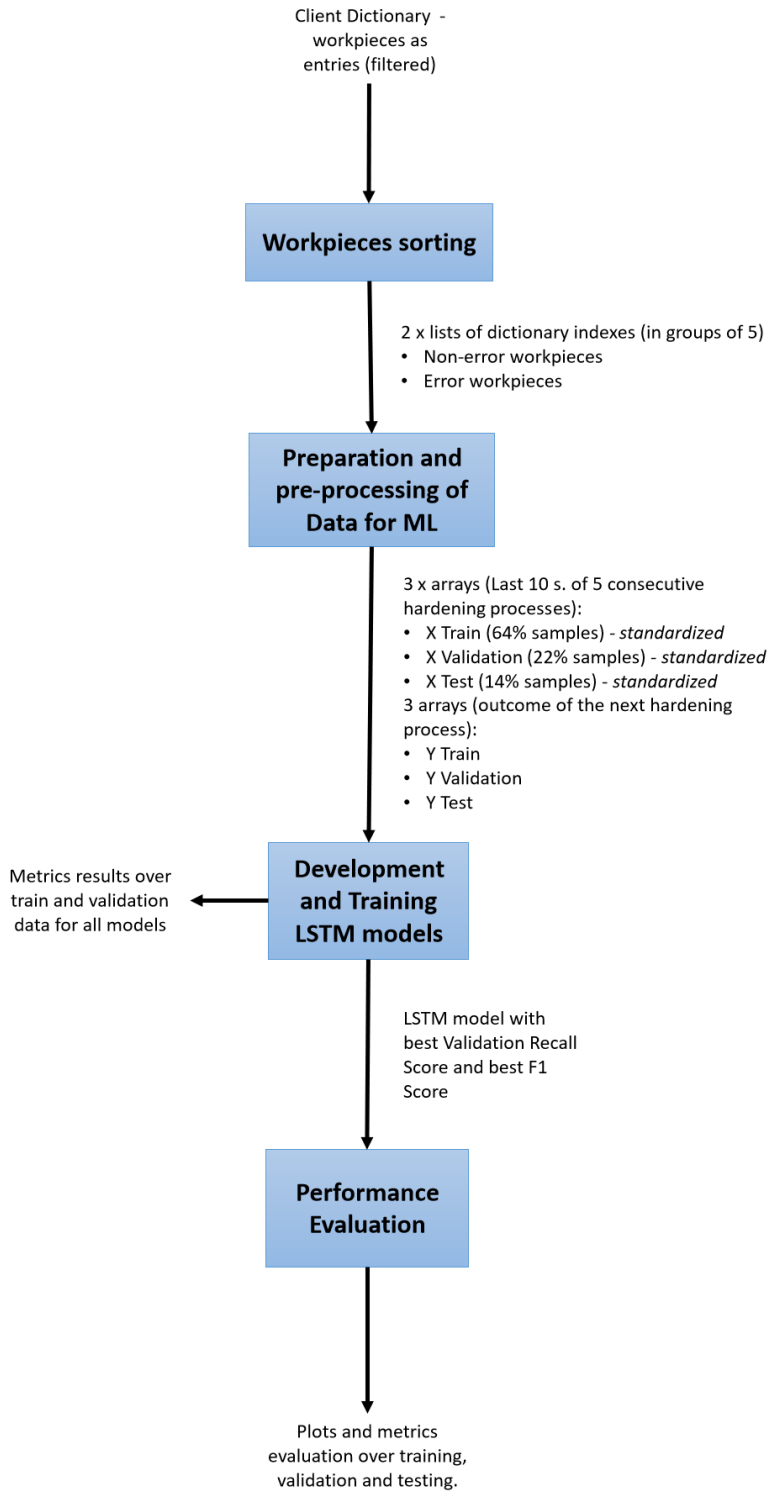


Figure 6.1: LSTM Case 2 Development - Simplified Block Diagram.

6.1 Data Selection

The development that follows from here until the end of the chapter, is present in the Jupyter notebook named *S6_1___S6_2___S6_3___S6_4_LSTM_Case2.ipynb*.

For this problem, the last 10 seconds of heating (50 samples) of the 5 previous heating cycles will be used to determine if the next piece to harden will fail due to a DC Voltage Too Low error. This will imply that the model inputs will be much larger, in that they will contain 5 groups of 50 samples with all the relevant variables. Therefore, to reduce some complexity of the problem, only the variables related to the converter and the counter (which is the time stamp within a hardening process) will be taken into consideration.

Therefore, the variables that will be kept for the LSTM NN development will be the following 6:

- Counter.
- Converter_Power.
- Converter_DC_Voltage.
- Converter_AC_Current.
- Converter_Frequency.
- Converter_Error.

In this problem, the timestamp of each one of the files is very important, since it is interesting to analyse consecutive workpieces to determine the outcome of the next one. Therefore, the first thing will be to sort the general dictionary containing all the work pieces' data sets of Client 2 by its keys. The keys correspond to the file name, which always starts with the time stamp that has the following format: *YYYY-MM-DD–hh-mm-ss-msmsms*.

Having this dictionary sorted, it is now important to identify the indices/placement of the relevant work pieces for this problem within the general dictionary. Therefore, three different lists will be created:

- Indices Error of Interest: Index/Placement of files within Job 4 that contain a DC Voltage Too Low Alarm.
- Indices No Errors: Index/Placement of files within Job 4 that do not contain any error.
- Indices Converter Off: Index/Placement of files within Client 2 where the converter is always off or the heating cycle is smaller than 10 seconds.

Having identified the position of the relevant files and of the files that are not possible to be used (the ones contained in Indices Converter off), these indices need to be grouped by 5 workpieces. To group the workpieces' indices, a different strategy is used for workpieces that have the error of interest and workpieces that do not have an error. For the error group, for each one of the indices present in the list, the previous 5 indices of the general Client 2 dictionary will be taken to form a group. If one of these 5 previous indices, is included in the Indices Converter off group, this will be skipped, and the previous one will be taken until having the 5 needed. Regarding the non-error group, 6 consecutive indices will be searched within the Indices No Errors list, discarding the last one and saving the first 5. This will ensure that when predicting a non-error occurrence, the 5 data sets used as input will also not contain any error and will be part of the same job within the client.

By doing this, the result is two lists containing 5-index groups:

1. Non-error indices group: 174.
2. Error of interest indices group: 8.

It is to consider, that when taking the 5 previous workpieces from an error file, some of them could belong to some of the workpieces present in the groups of non-error workpieces or even to another workpiece with an error (if the time difference between the error occurrences is very small). In fact, 910 indexes are used in total in this study case, but only 897 different workpieces are selected, meaning that there are 13 duplicated indexes spread in the different groups of the different data sets. In any case, there won't be any 5-index group of workpieces identical to another group (which are the ones used as inputs for the model), therefore this does not count as duplicated input data.

6.2 Data Preparation

The same procedure as for the previous model is followed regarding data splitting. Training, validation and testing data sets will be created from the available data with the same percentage distribution as for the previous study case. The difference is that the splitting is done over the index values, which are used in a later stage to import the corresponding data set of the workpiece. Therefore, 6 lists will be created:

- A training indices-list containing 122 non-error indices.
- A training indices-list containing 4 error indices.
- A validation indices-list containing 35 non-error indices.
- A validation indices-list containing 2 error indices.
- A testing indices-list containing 17 non-error indices.

- A testing indices-list containing 2 error indices.

Using those indices, the corresponding DataFrames containing the data set for the selected work pieces will be imported. For this case, the last 10 seconds while the converter is on, of the previous 5 work pieces will be used to determine if the next workpiece will fail or not. Therefore, for each pair of dictionaries mentioned above, a 4D array will be created. Once more, due to the unbalance of data, the error data will be duplicated 10 times. These three arrays have the following structure:

- First dimension: Number of groups containing 5 workpieces (162 training, 55 validation, 37 testing).
- Second dimension: Number of workpieces in each group (5).
- Third dimension: Number of samples of each workpiece / hardening process (50).
- Fourth dimension: Number of variables present in each workpiece / hardening process (6).

The corresponding boolean Y arrays will also be created using the first dimension of the corresponding X array.

The data will be again shuffled over the first dimension of the array, to avoid grouping of non-error files and the input data (X array) will be standardized.

The problem encountered here is that the LSTM model that will be trained with this data can only handle 3D arrays, where the first dimensions are the number of files/amount of samples used for training and the other 2 dimensions are the data set shape. Therefore, a packing and transformation of the data from a 4D array into a 3D array is needed. The easier and more correct way to do it is by merging the second and third dimensions into one. This will mean that instead of having 5 data sets of 50 samples each, the array will have a unique data set of 250 samples, which has the same meaning when training the model.

The data is now structured and pre-processed in the way needed. Training, validation and testing arrays with standardized data are randomly shuffled and ready to be used to design, train, validate and evaluate an LSTM NN model. These arrays have been saved and can be found in *LSTM_Files/LSTMVectors/Case2*

6.3 Model Development

In this case, the same type of model as the previous study will be used. A Sequential LSTM model that is configurable in the number of hidden layers, number of neurons per layer and activation function for the hidden layers will be created. It is still a binary classification problem, therefore the output will be a **Dense** layer with *Sigmoid* as the activation

function. The same loss metric (mean square error) and performance parameters (recall and precision) will be used in this case too.

Also in this case, 12 different models will be trained, but with different layers and neuron configurations. Due that the number of features extracted from the data sets is smaller, this will imply fewer neurons per layer. But at the same time, the overall amount of data used as input is bigger, therefore 2, 3 and 4 layers will be evaluated as the possible number of hidden layers in the model. The tested model combinations are the following:

- 2 layers: [4,2], [5,3].
- 3 layers: [4,3,2], [5,4,3].
- 4 layers: [4,3,3,2], [5,3,2,2].

In each layer, *ReLU* and *Sigmoid* will be again tested as possible activation functions.

Each one of these 12 models will be trained across epochs ranging from 50 to 200, with an interval of 50 epochs between each, and using batch sizes between 20 and 100, with an interval of 20 (as in the previous case). The 5-nested for loop method with evaluation on both recall and F1 score will be used here too.

Once again, the models are evaluated from smaller to higher complexity and size of training parameters. In this way, if two models have the same performance, the less complex model will be the one saved as the best model. The results obtained from this simulation are now shown in Table 6.1.

Table 6.1: Best Models Results - Case 2.

	Best F1 Score Model	**Best Recall Score Model**
Number of hidden layers	2	2
Number of neurons per layers	[5,3]	[4, 2]
Activation function in hidden layers	*ReLU*	*ReLU*
Output layer	1 Neuron - *Sigmoid*	1 Neuron - *Sigmoid*
Number of epochs	200	150
Batch size	80	100
Last Validation F1 Score	0.93	0.53
Last Validation Recall Score	1	1

The development and testing of this model can be executed by the reader using the file named *S6_3___S6_4_LSTM_Case2_ONLY_TrainingAndTesting.ipynb*, which takes as input parameter the previously saved vectors.

6.4 Model Performance Evaluation

All the figures present in this section, together with the dictionaries containing the models presented in the above table, can be found under *LSTM_Files/Results/Case2*.

Once again, observing the results in Table 6.1, both models end up with a validation recall of 1, therefore, the one with the higher F1 score will be selected to be evaluated against test data. In this case, both models have 2 layers, so even if the best F1 score model has a higher number of neurons and has been trained over 50 epochs more, the complexity difference between both models is almost irrelevant. The plots showing its performance evolution over the epochs are now shown in Figure 6.2.

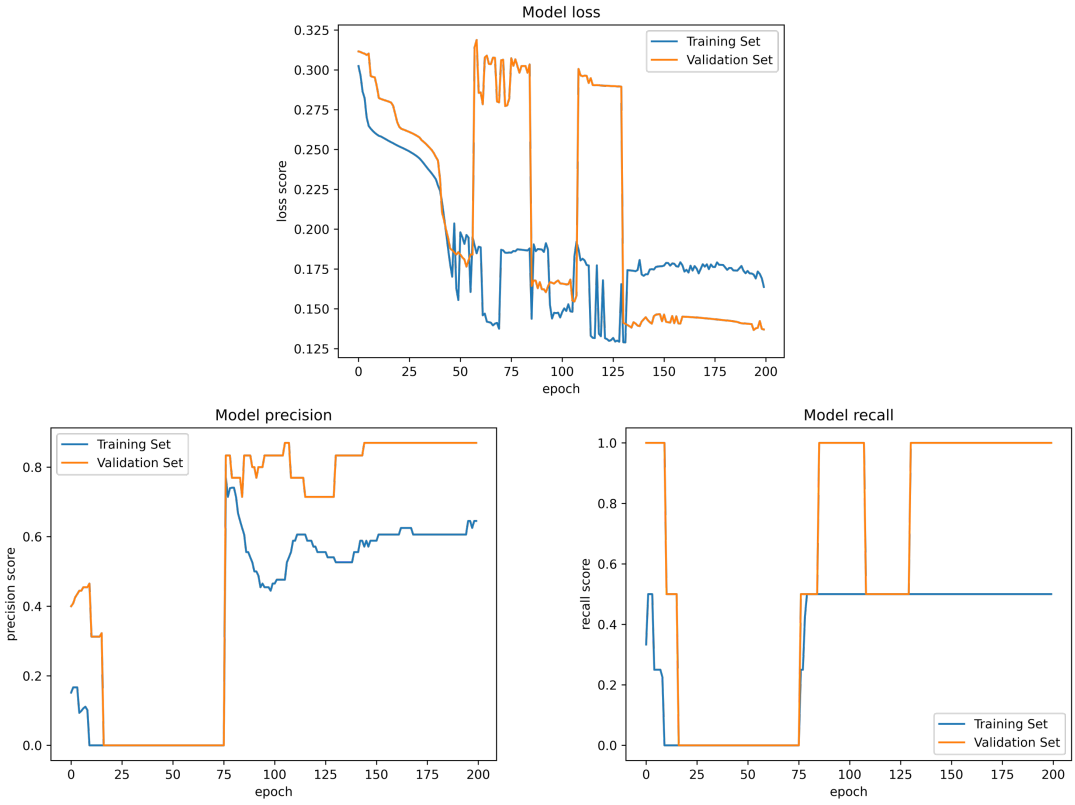


Figure 6.2: Performance Metrics of Best Model - Study Case 2.

Observing the model loss, this starts having a decreasing curve until around epoch 40. After that, the validation loss increases significantly and the model starts showing a repetitive behaviour until around epoch 130. During this period, the algorithm has difficulties in adjusting the model parameters to reduce the validation and training loss. This can be because, as explained before, in some of the groups of data, singular workpieces can be seen twice. After epoch 130, the loss stabilizes at around 0.175 for the training set and 0.13 for the validation set, which are low values and accepted as good results for this case. This shows that the complexity of the data found in the validation set is smaller or easier to understand for the final model than the one found in the training set.

Analysing the precision score, the behaviour of it in the validation and training set presents a quite similar behaviour. It starts by decreasing to 0 and stays there until epoch 75, meaning that for this period, no errors are correctly detected. After that, a sudden improvement is implemented in the model, increasing the precision of it in both data sets until around 0.8. From there on, the training precision decreases again until around 0.5, while the validation one maintains its good score. The algorithm adjusts the model parameters to improve this score on the training data set and achieves this by obtaining a final training precision score of around 0.65 and a validation precision score of approximately 0.85. Even if the precision score in the training data is not satisfactory, the one in the validation data is. Once again, this shows that the validation data is less complex or easier for the model to understand.

To conclude the performance metrics analysis, the recall score is observed. This metric starts with a very high value for the validation data set but with a not-so-high one for the training data set. In this case, as in the previous one, it is to take into consideration that the error files are duplicated ten times, but actually, there are only 4 types of errors in training and 2 in validation. This means that this score can only take values of 0, 0.25, 0.5, 0.75 and 1 for the training set, and of 0, 0.5 and 1 for the validation data set. This can be easily observed in the graphs, which have big jumps between these values along the training epochs. As for the precision graph, a 0 recall score is also shown in this graph until epoch 75. This is expected, due that both metrics have the number of true positives in their numerator, meaning that if no errors are correctly predicted, both metrics will be 0. After epoch 75, the model increases significantly its performance, being able to bring the final validation model recall to 1 (after variations due to the adjustments of different parameters along the epochs), but not the training one, which ends up being 0.5. The complexity and the amount of data in both data sets can be, one more time, the reason for this difference between the final scores.

Finally, to see if this model has a good or bad performance in a real-case scenario, this will be evaluated against the testing data and the value of the three metrics mentioned above will be calculated one more time. Also, in this case, the optimal threshold for the output signal is calculated and found to be 0.25. The results of the predictions given by the model over the test data can be seen in Figure 6.3.

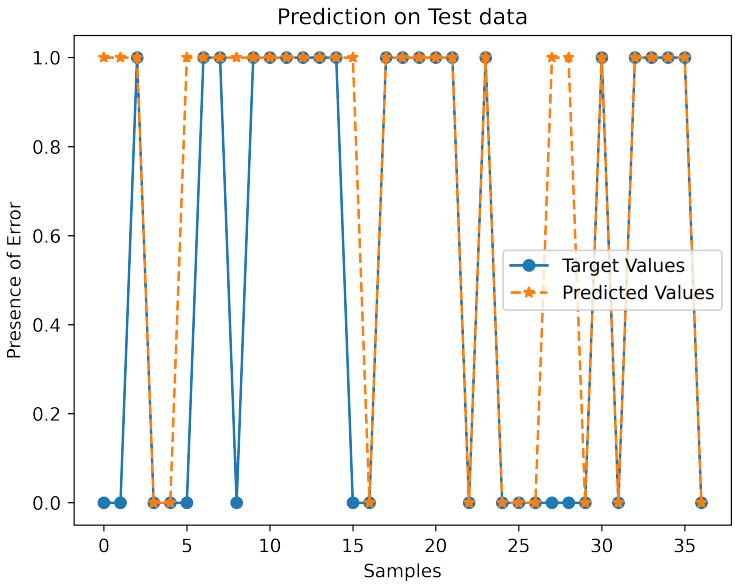


Figure 6.3: Model Predictions on test data - Study Case 2.

The model performance is very satisfactory, being able to identify correctly 30 samples of a total of 37, which corresponds to 81% of all the samples. The performance metrics over the testing data are presented in Table 6.2.

Table 6.2: Performance metrics over test data.

F1 Score	0.85
Precision	0.74
Recall	1

The recall score is also 1 over the testing data set, meaning that all the errors are correctly classified. Looking at the precision score, which is 0.74, means that some non-error groups are miss-classified as error groups. These two parameters give an overall performance, measured by the F1 score, of 0.85, which is a very good value. This means that the model works very well in real-world data and the goal of this study, which was to identify all the error groups, has been successfully achieved. It is to be noted that the high performance presented over the testing data set, could be because of the manually calculated threshold. When calculating performance metrics during the training of the model, a pre-defined threshold of 0.5 is internally used by the *Keras* function to output the metric value. It is observable that over the testing data, the best threshold found is 0.25, which is half the value of the predefined one. Changing this parameter in the function used for model training could present better results over the training and validation data sets.

7 Conclusions

After analysing the results obtained within the different topics studied in this project, several conclusions can be extracted. The first thing to point out is that this study has brought a big insight regarding the client data itself, and shown that things in theory are not the same as in practice. The selected clients for this project had the same type of machine, hoping that the data extracted from the three of them should at least be quite similar, due to the same heating process being performed (hardening). However, it has been found that singularities are not only among different clients but also within the same client (due to the different job configurations) and sometimes, even within the same job configuration. This has also shown that sometimes, ENRX designs certain parameters in its machines, with the idea that they should be used in a certain way, but possibilities exist that these parameters are not used as expected sometimes (as for the job configuration parameter in Client 2 or the Energy Alarms). This points out that it is not the same to analyse process data for a process company that has a production line which is always doing the same job, as for a company, like ENRX, which delivers products that can be used for many different processes in many different ways. ENRX hoped to find a way to create a system that could be suitable for each one of the machines sold. After this study, it was found that the solution delivered should be probably customer-oriented and not machine-oriented, since the same machine in two different customers could have a completely different data set behaviour.

If offering a customized solution for each customer, a previous study of how the machine works in that specific production line, how the customer uses the machine and which variables are relevant, should be done before starting the collection of data. It is vital to ensure that the system behaves as expected when using a data-driven model for predictive maintenance. This means that the customers should also use their systems in a more uniform way to ensure good results.

As shown in the Exploratory Data Analysis, many of the collected variables did not have any relevance for the study, while others were only relevant for some of the customers. If the data is intended to be collected in the cloud over several years, it is important to not collect unnecessary data. The amount of data to collect depends not only on the number of variables collected but also on the sampling frequency of them. The used sampling rate is 200 ms., giving quite good results for the problem studied, but this would depend on the aim of the study itself. Process variables from the frequency converter change much faster than the variables from the axis machine, therefore a balance should be found for

every single problem or model to develop. The problem of the amount of data and its sampling is a very well-discussed topic in Section 5.2 and Chapter 6 of the article named 'Anticipating Future Behavior of an Industrial Press Using LSTM Networks'[9]. In the problem under study in that article, functioning data of an industrial process over 2.5 years have been collected with a sampling rate of 1004 samples per day, ending up with a data set containing almost 1.5 million data points. To accelerate the LSTM training, the data set was reduced to one and two samples per day (by averaging the 1004 samples collected), showing that (even with a slightly worse performance) the model was able to predict quite well some of the failures wanted. This shows that sometimes a compromise between the amount of data, the processing speed and the results achieved must be taken.

In this decision, it is to be taken into account that some of the customers will be sending this data to the cloud using the mobile network, which can have limitations in the amount of data that can be sent at the same time and how much time does that take (to determine when the next sample can be). In addition, previous data filtering needs to be performed. As shown during the study, some of the data sets resulted in being very short or with no power on at all. These data sets should probably not be stored, because they do not show the real behaviour of the machine during production.

As explained, the data that needs to be collected is also very dependent on what needs to be found. Different problems may require different types of variables that can or cannot be present in the actual system. An example could be if wanted to determine process efficiency, the temperature generated in the workpiece would be interesting to have. Unfortunately, this variable is not present in the actual system, and due to the high temperature at which the machine can operate, it would also not be that easy to implement the right sensor that would give an accurate measurement just on the heated area. It is also to remark that when analysing the hardening process itself, additional information regarding the piece under hardening, the coil used for hardening, the hardening result, etc. needs to be collected. In the actual system, this information is not available from the machine itself, and the customers (even if asked for it) have not shared this information for this project. This makes it difficult to analyse the hardening process itself because no feedback from it is given.

The main idea is that the final implementation of this system will be on a cloud server, where the data is collected and analysed. This is an optimal solution, due to the possibility of having "unlimited" space and high computational power to analyse the data, for example, with ML models. But it is to remember that this cloud system, does not and will not have any direct control on the machine itself. Therefore, if thinking of an error predictive model, the cloud server should be able to communicate with the in-place system. Thinking of the second LSTM NN built into this project (which can detect if the next workpiece will fail), a signal with this output should be sent to the HMI present in the machine, to indicate to the operator about this possible outcome and expect him to take actions for it. This communication protocol will vary according to which type of

model is used, the errors predicted, etc. However, it is to point out that if a data-driven model is built with the expectations of solving real-time problems, this could be very difficult. It is to account that things happen very fast in the system, and by the time the data is sent to the server, processed and the result is sent back, many things could have happened in the system already. Therefore, if a ML algorithm is found interesting to use in real-time, the possibility to add more computational power to the offline system should be evaluated.

Regarding the development of the LSTM Neural Network models has shown the importance of analysing different models, both in internal complexity and training parameters, to determine the one that will give an overall better behaviour. It has also been shown the importance of selecting the right threshold when dealing with classification problems. Another important fact that has been seen is the importance of error data duplication when dealing with highly unbalanced data sets. Even if doing this will imply having step-values for some of the performance metrics (like the recall score), the duplication of data gives the possibility for the model to learn from it. If error data is not duplicated, and instead of having 80 total samples, only 8 are available, when the algorithm tries to fit the data to the model, it will interpret these files as outliers or anomalous data, and it will fit the model only to the non-error data. It is also to point out that the second model has given higher results over testing data, despite its higher complexity. This could maybe be because of the variables selected as inputs. In case 2, only converter variables are used as input in the model, which could maybe be more relevant to determine a converter error, than when taking also into consideration some variables from the axis machine.

To conclude, it is to point out that the results obtained in this project are very satisfactory since they prove that the data that can be collected from the machines has some meaningful information and that there is place for a further development with it. Regardless of the singularities, if reducing the complexity of the problem to solve, good results are achieved, giving space for improvement and development over different and more complex problems. It is also to point out that even if many different Machine learning methods can be tested for different purposes, the use of LSTM NN has been very satisfactory, due to its ability to correctly identify patterns in time-series data. Therefore, it is a good starting point for this first feasibility study of the usability and possibilities within Information Services that will maybe be offered in the future by ENRX.

7.1 Future work

As stated, this study is just a first look into the possibilities that can be developed to analyse and use customer data from ENRX's machines, therefore many improvements and paths to move forward are possible. Focusing first on the data collection itself, it would be relevant to collect not only customer data, which gives an overview of real

working conditions but also laboratory data. This means having the machine under study available for testing and collecting more data related to failures, anomalies, etc. This would give a more varied data set, that could give the possibility to extract other relevant information or analyse other types of problems. In addition, this could also give the possibility to test new sensors that could have relevance in the system. An example of this is shown in Section 5.1 of the article 'LSTM-Based Stacked Autoencoders for Early Anomaly Detection in Induction Heating Systems'[8]. In the mentioned article, training and testing data is collected on a real system in a laboratory, giving the possibility to obtain a very varied data set with all the possible failures well documented.

If continuing with the development of LSTM NN for error prediction, it would be interesting to analyse how good these models are in predicting other types of errors, and how much is the minimum amount of data needed to detect each one of the errors correctly. This will give an overview of how many or which type of errors are these systems able to detect from the available data, giving an idea of what is the company able to provide to the customer.

To assure good results, the usage of the machine should match the criteria used to build data-driven models. Therefore, procedures that the customers should follow to assure good prediction results as well as appropriate machine setups should be also developed.

Another Machine Learning method that would be interesting to implement is an unsupervised ML algorithm over the Energy Alarms. As described in Section 5.1, many of the energy alarms collected from the customers are not real, they just show a wrong setting within the machine. An unsupervised ML model could be able to differentiate this data into two groups: workpieces that have an energy alarm, and workpieces that don't have an energy alarm but the machine limit is wrong configured. This could help in having an overview of how many pieces are discarded by the customer because of this error. Different methods within clustering such as k-means, Gaussian Mixture Model, etc., could be tested.

Going away from the ML analysis when having the possibility to collect a large amount of data regarding working conditions, different statistical parameters could also be calculated and displayed in graphical forms. Examples can be: a histogram of the converter errors that occurred per machine/job, statistical variables regarding converter process variables (such as mean, min, max and std of the AC current delivered by the converter), more frequently used job (with its configurations), etc. This will give a more detailed overview both to ENRX and to the customer of how the machine is used, creating a whole new world of possibilities regarding efficiency improvement, design of tailored solutions, etc.

The data is available and the possibility to analyse it has been shown, therefore many studies can be now carried out on it according to customer's and company's needs.

References

[1] McKinsey and Company, 'What are industry 4.0, the fourth industrial revolution, and 4ir?,' Aug. 2022. [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir#/ (visited on 26/10/2023).

[2] IBM, 'What is industry 4.0?,' [Online]. Available: https://www.ibm.com/topics/industry-4-0 (visited on 26/10/2023).

[3] 'Enrx.' (2023), [Online]. Available: https://www.enrx.com/ (visited on 26/10/2023).

[4] ENRX. 'Induction heating products.' (2023), [Online]. Available: https://www.enrx.com/en/Induction-Products/Induction-heating-equipment (visited on 26/10/2023).

[5] ENRX. 'Induction heating applications.' (2023), [Online]. Available: https://www.enrx.com/en/Induction-Applications/Induction-heating-applications (visited on 26/10/2023).

[6] Z. Ge, Z. Song, S. X. Ding and B. Huang, 'Data mining and analytics in the process industry: The role of machine learning,' *IEEE Access*, vol. 5, pp. 20 590–20 616, 2017. DOI: 10.1109/ACCESS.2017.2756872.

[7] J. Talar, 'Data mining methods-application in metallurgy,' *Archives of Metallurgy and Materials*, vol. 52, no. 2, 2007.

[8] M. Qais, S. Kewat, K. Loo, C.-M. Lai and A. Leung, 'Lstm-based stacked autoencoders for early anomaly detection in induction heating systems,' *Mathematics*, vol. 11, p. 3319, Jul. 2023. DOI: 10.3390/math11153319.

[9] B. C. Mateus, M. Mendes, J. T. Farinha and A. M. Cardoso, 'Anticipating future behavior of an industrial press using lstm networks,' *Applied Sciences*, vol. 11, no. 13, 2021, ISSN: 2076-3417. DOI: 10.3390/app11136101.

[10] S. F. Panzeri Reyes. 'Data analysis and modelling of induction hardening processes - gitlab repository.' (2023), [Online]. Available: https://gitlab.com/panzeristefano99/data-analysis-and-modelling-of-induction-hardening-processes (visited on 28/10/2023).

[11] Wikipedia, 'John tukey,' Sep. 2023. [Online]. Available: https://en.wikipedia.org/wiki/John_Tukey (visited on 26/10/2023).

[12] Wikipedia, 'Data science,' Oct. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Data_science (visited on 26/10/2023).

[13] J. Frankenfield, 'Data analytics: What it is, how it's used, and 4 basic techniques,' Aug. 2023. [Online]. Available: https://www.investopedia.com/terms/d/data-analytics.asp (visited on 26/10/2023).

[14] IBM, 'What is machine learning?,' [Online]. Available: https://www.ibm.com/topics/machine-learning (visited on 26/10/2023).

[15] Wikipedia, 'Machine learning,' Oct. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning (visited on 26/10/2023).

[16] Wikipedia, 'Curse of dimensionality,' Sep. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Curse_of_dimensionality (visited on 26/10/2023).

[17] K. Soni, 'Machine learning being aggressively adopted by many sectors,' Dec. 2021. [Online]. Available: https://cionews.co.in/machine-learning-being-aggressively-adopted/ (visited on 26/10/2023).

[18] IBM, 'What are neural networks?,' [Online]. Available: https://www.ibm.com/topics/neural-networks (visited on 26/10/2023).

[19] Wikipedia, 'Neural network,' Oct. 2023. [Online]. Available: https://en.wikipedia.org/wiki/Neural_network (visited on 26/10/2023).

[20] harkiran78, 'Artificial neural networks and its applications,' Jun. 2023. [Online]. Available: https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/ (visited on 26/10/2023).

[21] Great Learning Team, 'Types of neural networks and definition of neural network,' Nov. 2022. [Online]. Available: https://www.mygreatlearning.com/blog/types-of-neural-networks/ (visited on 26/10/2023).

[22] P. Narasimman, '9 types of neural networks: Applications, pros, and cons,' Sep. 2023. [Online]. Available: https://www.knowledgehut.com/blog/data-science/types-of-neural-networks (visited on 26/10/2023).

[23] IntelliPaat, 'What is lstm? introduction to long short term memory,' Sep. 2023. [Online]. Available: https://intellipaat.com/blog/what-is-lstm/ (visited on 26/10/2023).

[24] aakarsha chugh, 'Deep learning | introduction to long short term memory,' May 2023. [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (visited on 26/10/2023).

[25] C. Olah, 'Understanding lstm networks,' Aug. 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 26/10/2023).

[26] S. Dobilas, 'Lstm recurrent neural networks — how to teach a network to remember the past,' Feb. 2022. [Online]. Available: https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e (visited on 26/10/2023).

[27] ENRX. 'Induction heating - how it works.' (2023), [Online]. Available: https://www.virtualexperience.enrx.com/#/en/applications/howitworks (visited on 26/10/2023).

[28] Ambrell, *About induction heating*, Brochure, 2022. [Online]. Available: https://www.ambrell.com/hubfs/Ambrell_PDFs/411-0169-10.pdf (visited on 26/10/2023).

[29] BYJU'S, 'What are eddy currents?,' [Online]. Available: https://byjus.com/physics/what-are-eddy-currents/ (visited on 26/10/2023).

[30] ENRX. 'Induction hardening.' (2023), [Online]. Available: https://www.virtualexperience.enrx.com/#/en/applications/hardeningTempering (visited on 26/10/2023).

[31] P. Karia, '4 types of metal hardening,' Jan. 2023. [Online]. Available: https://blog.thepipingmart.com/metals/4-types-of-metal-hardening/ (visited on 26/10/2023).

[32] Metals Engineering. 'Hardening.' (2020), [Online]. Available: https://metalsengineering.com/hardening/ (visited on 26/10/2023).

[33] ENRX. 'Hardline: Induction hardening machines.' (2023), [Online]. Available: https://www.enrx.com/en/Induction-Products/Induction-heating-equipment/Hardline (visited on 26/10/2023).

[34] ENRX. 'Hardline m - modular induction hardening machine.' (Nov. 2021), [Online]. Available: https://www.youtube.com/watch?v=wi5Dvx9BskM (visited on 26/10/2023).

[35] 'Ydata profiling.' (2023), [Online]. Available: https://docs.profiling.ydata.ai/4.6/ (visited on 26/10/2023).

[36] 'Seaborn.pairplot.' (2023), [Online]. Available: https://seaborn.pydata.org/generated/seaborn.pairplot.html (visited on 26/10/2023).

[37] B. U. S. of Public Health, 'The correlation coefficient (r),' Apr. 2021. [Online]. Available: https://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717-Module9-Correlation-Regression/PH717-Module9-Correlation-Regression4.html (visited on 26/10/2023).

[38] J. Brownlee. 'Machine learning mastery.' (2023), [Online]. Available: https://machinelearningmastery.com/ (visited on 26/10/2023).

[39] J. Brownlee, 'How to develop lstm models for time series forecasting,' Aug. 2020. [Online]. Available: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/ (visited on 26/10/2023).

[40] J. Werth, 'Lstm for predictive maintenance on pump sensor data,' Aug. 2021. [Online]. Available: https://towardsdatascience.com/lstm-for-predictive-maintenance-on-pump-sensor-data-b43486eb3210 (visited on 26/10/2023).

[41] I. Valchanov, 'Obtaining standard normal distribution step-by-step,' Apr. 2023. [Online]. Available: https://365datascience.com/tutorials/statistics-tutorials/standardization/ (visited on 26/10/2023).

[42] F. Chollet, 'The sequential model,' Apr. 2020. [Online]. Available: https://keras.io/guides/sequential_model/ (visited on 26/10/2023).

[43] Naveen, 'What is relu and sigmoid activation function?,' Apr. 2022. [Online]. Available: https://www.nomidl.com/deep-learning/what-is-relu-and-sigmoid-activation-function/ (visited on 26/10/2023).

[44] J. Brownlee, 'A gentle introduction to the rectified linear unit (relu),' Aug. 2020. [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/ (visited on 26/10/2023).

[45] T. Jacob, 'Vanishing gradient problem: Causes, consequences, and solutions,' Jun. 2023. [Online]. Available: https://www.kdnuggets.com/2022/02/vanishing-gradient-problem.html (visited on 26/10/2023).

[46] 'Sklearn.model_selection.gridsearchcv.' (2023), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (visited on 26/10/2023).

Appendix A

Master Thesis Description

In this appendix, the document containing the initial Master's Thesis Description, which has been used as a guideline to fulfil this project is found. This document was written by the author of this project before the start of it, and approved by the representatives of both the company (ENRX AS) and the university (University of South-Easter Norway, Campus Porsgrunn).

University of South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: Data Analysis and Modelling of Induction Hardening Processes

USN supervisor:

- Main Supervisor: Håkon Viumdal
- Co-Supervisor: Ru Yan

External partner: ENRX (former EFD Induction)

- Technical Co-Supervisor: Audun Hystad
- Progress Co-Supervisor/Contact Person: Bjørn Sverre Aspheim

Task background:

ENRX is working on a project to provide remote service to their customers. In a very brief way, this system consists of a router that provides remote access (through a VPN tunnel) to the control system of the machine. In this way, a service technician can remotely access the machine, having access to real time information and full control of it.
This project is divided into two parts: “Remote Services” (explained above) and “Information Services”. The last part consists of collecting the functioning data of the machine and store it in a cloud data base for analysis. The idea is to see if analysing the functioning data of the machine can help in predicting future failures, more efficient working patterns, etc.

Task description:

Functioning data from different customers machines need to be remotely collected and analysed using Multivariate Data Analysis/Machine Learning (ML) methods. The goal is to find patterns that can describe the process and machine behaviour.

To achieve the main objective, the following subtasks are to be accomplished:

Introduction:

- Principal problem description and fundamental goal description of the project in general terms.
- Outline the prevailing induction heating process at ENRX, including basics about the working principle, applications, and ENRX’s machines.
- Outline the prevailing hardening process, including basics about how the process works and small description of induction heating hardening process.
- Carry out a literature review regarding data collection and/or analysis of induction heating processes or processes with similar dynamics.

System description:

- Describe the process(es) and machine(s) under scrutiny: More detailed description of how the ENRX hardening machine works and description of its specifications.

- Describe the data collection system: Described the development and implementation of the data collection system on customer's machines. Explanation of how it works and overview of the collected variables.

Data Analysis:

- Execute exploratory analysis on sampled data from the system: Correlation analysis between variables, relevant plotting (e.g., histogram plot or scatter matrix plot), etc.
- Perform pre-processing of the collected data, in order to be utilized for machine learning approaches (e.g., structuring and filtration of data).
- Define, develop, and validate one or more pertinent data-driven models for the given application.

Discussions:

- Discuss the data collection method: Discussion about collecting frequency, amount of data, data format, etc.
- Discuss the quality of the data collected: Discussion about if the available data is sufficient to get relevant information about the process and/or machine.
- Discuss how the data-driven models could be implemented and operate in the system.

Student category: Reserved for the IIA – Industry Master student Stefano Fernando Panzeri Reyes

Is the task suitable for online students (not present at the campus)? No

Practical arrangements:
Necessary HW and SW provided by the company.

Supervision:
As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:

Date and Place: Skien, 23 June 2023.

USN Main Supervisor: HÅKON VIUMDAL

ENRX Technical Supervisor: AUDUN HYSTAD

Student: STEFANO FERNANDO PANZERI REYES

Appendix B

Frequency Converter's Nomenclature

Usually the converters have two numbers and two letters. The numbers indicate at which power they can be run continuously (the first number, always lower) and intermittently (the second number, always higher). The first letter indicates whether it is a serial (S) or parallel (P) converter. The second one indicates the frequency range which is designed for: low (L), medium (M) or high (H). Some machines can operate in different frequency ranges by short-cutting one of the output capacitors with a parallel switch. In case these two frequencies are not inside the same level range, the two numbers at the beginning of the name correspond to the max intermittent power in each one of the ranges and an additional frequency letter is added to indicate the two ranges.

Appendix C

Collected variables from machines

Table C.1: Variables Collected.

Name	Type	Size (bytes)	Units	Comments
Counter	Int	2	-	Counter for messages sent
TimeStamp	String[30]	32	-	
NewPiece	Bool	1	-	
Automatic	Bool	1	-	True = automatic mode; False = MakeReady mode
DummyByte	Byte	1	-	To complete one word sending
NC_Active	Bool	1	-	NC program is running
JobName	String[30]	32	-	Name of NC program
C_AxisMode	Bool	1	-	True = Rotary table; False = Spindle
RobotInUse	Bool	1	-	
FrequencySelected	Int	2	-	2 = low frequency; 1 = high frequency ; 0 = not selected
LubricationAcitve	Bool	1	-	
WP_IO	Bool	1	-	
WP_NIO	Bool	1	-	
MachineIsEmpty	Bool	1	-	Machine has run empty (no piece)
HeatingOn	Bool	1	-	Converter heating is on
HeatingCycleComplete	Bool	1	-	
EnergySetpoint_min	Real	4	$impulsesorkW$	
EnergySetpoint_max	Real	4	$impulsesorkW$	
Inductor_WaterFlow_min	Real	4	l/min	
Inductor_WaterFlow_max	Real	4	l/min	
Inductor_WaterTemp_min	Real	4	$°C$	
Inductor_WaterTemp_max	Real	4	$°C$	
BusBar_WaterFlow_min	Real	4	l/min	
BusBar_WaterFlow_max	Real	4	l/min	
BusBar_WaterTemp_min	Real	4	$°C$	
BusBar_WaterTemp_max	Real	4	$°C$	
Quench1_WaterFlow_min	Real	4	l/min	
Quench1_WaterFlow_max	Real	4	l/min	
Quench2_WaterFlow_min	Real	4	l/min	
Quench2_WaterFlow_max	Real	4	l/min	
Quench1_WaterTemp_min	Real	4	$°C$	
Quench1_WaterTemp_max	Real	4	$°C$	
Quench2_WaterTemp_min	Real	4	$°C$	
Quench2_WaterTemp_max	Real	4	$°C$	
Converter_Setpoint	Real	4	%	
Converter_Power	Real	4	kW	
Converter_DC_Voltage	Real	4	V	
Converter_AC_Current	Real	4	A	
Converter_Frequency	Real	4	kHz	

Converter_WaterFlow1	Real	4	-	
Converter_WaterFlow2	Real	4	-	
Converter_WaterTemp	Real	4	$°C$	
X_Axis_Acceleration	Real	4	mm/min^2	
Y_Axis_Acceleration	Real	4	mm/min^2	
Z_Axis_Acceleration	Real	4	mm/min^2	
C_Axis_Acceleration	Real	4	mm/min^2	
X_Axis_Velocity	Real	4	mm/min	
X_Axis_Position	Real	4	mm	
Y_Axis_Velocity	Real	4	mm/min	
Y_Axis_Position	Real	4	mm	
Z_Axis_Velocity	Real	4	mm/min	
Z_Axis_Position	Real	4	mm	
C_Axis_Velocity	Real	4	$°/min$	
C_Axis_Position	Real	4	$°$	
Inductor_WaterFlow	Real	4	l/min	
Inductor_Temperature	Real	4	$°C$	
BusBar_WaterFlow	Real	4	l/min	
BusBar_Temperature	Real	4	$°C$	
Quench1_WaterFlow	Real	4	l/min	
Quench1_Temperature	Real	4	$°C$	
Quench2_WaterFlow	Real	4	l/min	
Quench2_Temperature	Real	4	$°C$	
EnergyAlarm_Low	Bool	1	-	
EnergyAlarm_High	Bool	1	-	
Converter_Error	FC_error	4	-	
X_Axis_Error	PLC_error	2	-	
X_Axis_Warning	PLC_error	2	-	
Y_Axis_Error	PLC_error	2	-	
Y_Axis_Warning	PLC_error	2	-	
Z_Axis_Error	PLC_error	2	-	
Z_Axis_Warning	PLC_error	2	-	
C_Axis_Error	PLC_error	2	-	
C_Axis_Warning	PLC_error	2	-	

Appendix D

Evaluation of feature's relevances - Working Table

The table that is present in this appendix is the Excel table used to evaluate the relevance of the collected variables to filter the data set for further analysis.

Variables	Client 1 - Sinac 100/160 SM	Client 2 - Sinac150/70 SMH	Client 3 - Sinac 100 PM	ACTIONS
Counter				Keep Variable
TimeStamp				Keep Variable
NewPiece				Remove Variable
Automatic	<0.1 % Manual	0.1 % Manual	<0.1 % Manual	Remove Variable --> Remove CSV with Manual Heating Cycles
NC_Active	NO CHANGE -->	NO CHANGE -->	NO CHANGE -->	Remove Variable
JobName				Keep Variable
C_AxisMode	NO CHANGE --> Always Spindle	62.5% Spindle and 37.5% Rotary	NO CHANGE --> Always Spindle	Keep Variable
RobotInUse	NO CHANGE -->	NO CHANGE -->	NO CHANGE -->	Remove Variable
FrequencySelected	2% Low instead of High	NO CHANGE --> Not selected	NO CHANGE --> Not selected	Remove Variable
LubricationAcitve	NO CHANGE -->	NO CHANGE -->	NO CHANGE -->	Remove Variable
WP_IO	NO CHANGE --> Not working	NO CHANGE --> Not working	NO CHANGE --> Not working	Remove Variable
WP_NIO	NO CHANGE --> Not working	NO CHANGE --> Not working	NO CHANGE --> Not working	Remove Variable
MachineIsEmpty	NO CHANGE --> Always 0	<0.1 % Ran empty	<0.1 % Ran empty	Remove Variable --> Remove CSV with Empty Heating Cycles
HeatingOn				Keep Variable
HeatingCycleComplete	NO CHANGE --> Not working	NO CHANGE --> Not working	NO CHANGE --> Not working	Remove Variable
EnergySetpoint_min				Remove Limits for now
EnergySetpoint_max				
Inductor_WaterFlow_min	NO CHANGE		NO CHANGE	
Inductor_WaterFlow_max	<0.1 % 75			
Inductor_WaterTemp_min	NO CHANGE	NO CHANGE	NO CHANGE	
Inductor_WaterTemp_max	NO CHANGE	NO CHANGE	NO CHANGE	
BusBar_WaterFlow_min	NO CHANGE	NO CHANGE	NO CHANGE	
BusBar_WaterFlow_max	NO CHANGE	NO CHANGE	NO CHANGE	
BusBar_WaterTemp_min	NO CHANGE	NO CHANGE	NO CHANGE	
BusBar_WaterTemp_max	NO CHANGE	NO CHANGE	NO CHANGE	
Quench1_WaterFlow_min				
Quench1_WaterFlow_max				
Quench2_WaterFlow_min	NO CHANGE		NO CHANGE	
Quench2_WaterFlow_max			NO CHANGE	
Quench1_WaterTemp_min	NO CHANGE	NO CHANGE	NO CHANGE	
Quench1_WaterTemp_max	NO CHANGE	NO CHANGE	NO CHANGE	
Quench2_WaterTemp_min	NO CHANGE	NO CHANGE	NO CHANGE	
Quench2_WaterTemp_max	NO CHANGE	NO CHANGE	NO CHANGE	
Converter_Setpoint				Keep Variable
Converter_Power				Keep Variable --> Set to 0 frequency when AC Current is OFF
Converter_DC_Voltage				Keep Variable
Converter_AC_Current				Keep Variable
Converter_Frequency				Keep Variable --> Set to 0 frequency when AC Current is OFF
Converter_WaterFlow1		NO CHANGE --> Not working	NO CHANGE --> Not working	Keep Variable where present
Converter_WaterFlow2		NO CHANGE --> Not working	NO CHANGE --> Not working	Keep Variable where present
Converter_WaterTemp	NO CHANGE --> Not working	NO CHANGE --> Not working	NO CHANGE --> Not working	Remove Variable

X_Axis_Acceleration	NO CHANGE		NO CHANGE	Remove Variable (X axis not in use)
Y_Axis_Acceleration				Keep Variable
Z_Axis_Acceleration				Keep Variable
C_Axis_Acceleration			NO CHANGE	Keep Variable
X_Axis_Velocity	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable (X axis not in use)
X_Axis_Position	NO CHANGE	Different positions (manually adjusted) but constant during heating cycle	NO CHANGE	Remove Variable (X axis not in use)
Y_Axis_Velocity				Keep Variable
Y_Axis_Position				Keep Variable
Z_Axis_Velocity				Keep Variable
Z_Axis_Position				Keep Variable
C_Axis_Velocity			NO CHANGE	Keep Variable
C_Axis_Position			NO CHANGE	Keep Variable
Inductor_WaterFlow				Keep Variable
Inductor_Temperature				Keep Variable
BusBar_WaterFlow				Keep Variable
BusBar_Temperature				Keep Variable
Quench1_WaterFlow		NO CHANGE -->		Keep Variable where present
Quench1_Temperature		NO CHANGE -->		Keep Variable where present
Quench2_WaterFlow		NO CHANGE --> Not working (ext. Cooling station)	NO CHANGE --> Not present	Keep Variable where present
Quench2_Temperature		NO CHANGE --> Not working (ext. Cooling station)	NO CHANGE --> Not present	Keep Variable where present
EnergyAlarm_Low			NO CHANGE	Keep Variable
EnergyAlarm_High	NO CHANGE		NO CHANGE	Keep Variable
Converter_Error				Keep Variable
X_Axis_Error	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable (X axis not in
X_Axis_Warning	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable (X axis not in
Y_Axis_Error	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable
Y_Axis_Warning	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable
Z_Axis_Error	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable
Z_Axis_Warning	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable
C_Axis_Error	NO CHANGE	NO CHANGE	NO CHANGE	Remove Variable
C_Axis_Warning	86%	NO CHANGE	NO CHANGE	Keep Variable

Appendix E

Client's Correlation Analysis - Tables and Graphs

In this appendix, which is in A3 landscape format, the Excel tables containing all the strong linear correlation values found for each client with their respective comments are found. In addition, for each client, three scatter matrices that show the behaviour of each pair of variables (to detect non-linear correlations) are also shown. The three matrices correspond to the three working modes analysed: ALL data, converter ON data and converter OFF data.

The table containing the strong common correlations for the three clients will also be attached at the end. This table is already presented in Table 4.2, but here it will be possible to see it in a bigger format.

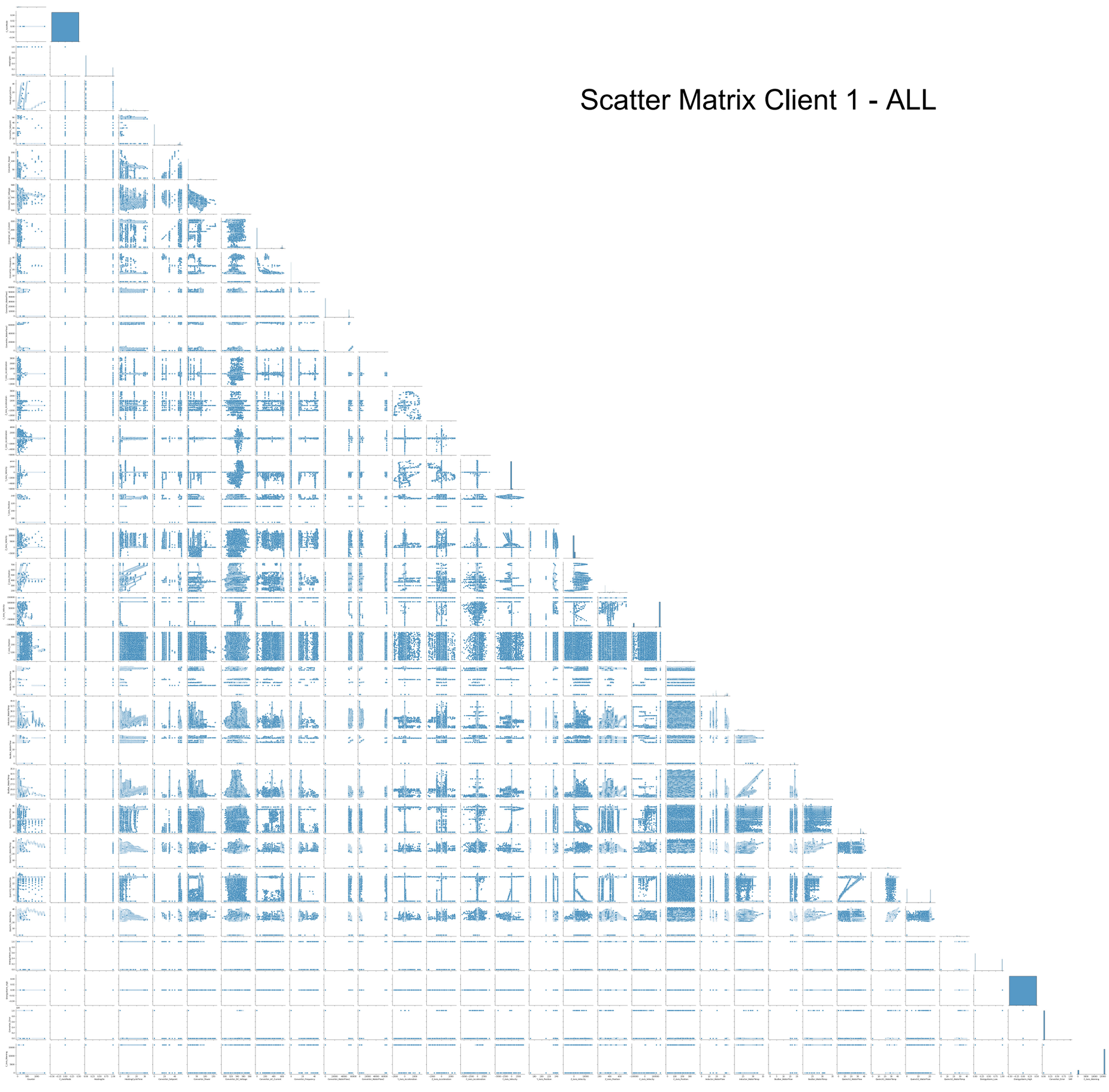
E.1 Correlations Client 1

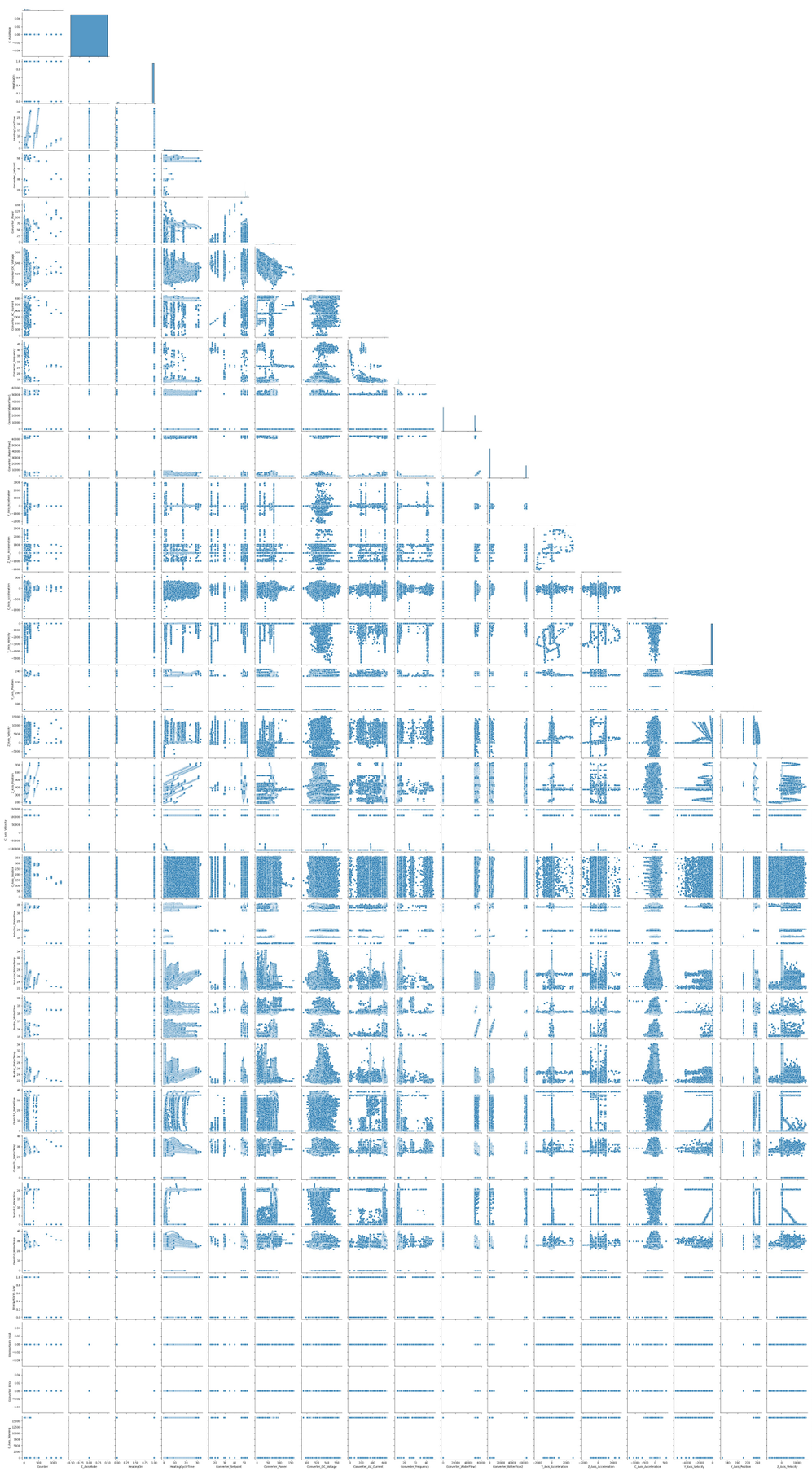
Client 1					Comments		
Variables		Correlations					
Variable 1	Variable 2	ALL	ON	OFF	Observations	Seen in Scatter Matrix?	
Converter_Setpoint	Converter_AC_Current	0.9988	0.9684	-	Setpoint and output current behave in the same way. Only present if Converter is ON	Yes	
C_Axis_Velocity	C_Axis_Warning	0.9958	0.9959	0.9958	A warning in the C Axis will affect the behaviour of its movement	Yes. No negative (or very little) speed with C Axis Warning ON	
Converter_Setpoint	Converter_Power	0.9823	0.5386	-	Setpoint and Power turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Yes	
Converter_Power	Converter_AC_Current	0.9817	0.5321	-	Power and current behave in the same way.	Not so clear	
Inductor_WaterFlow	C_Axis_Warning	0.9654	0.9674	0.9646	No Explanation Found	Yes. Bigger starting values of Wateflow when C Axis Warning ON.	
JobName	C_Axis_Warning	0.9645	0.9676	0.9632	The C_Axis_Warning does not appear in all jobs, but in some of them in different proportions. Can be related to the job itself or to when that job was executed.	JobName not shown in Scatter Matrix	
Y_Axis_Position	C_Axis_Warning	0.9638	0.9674	0.9624	No Explanation Found	Yes. Bigger starting values of Y Axis Position when C Axis Warning ON.	
Converter_AC_Current	HeatingOn	0.9592	0.2610	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	Because of Time delay between PLC signal and actual OFF of converter
Converter_Frequency	HeatingOn	0.9582	0.1999	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	
Converter_Setpoint	HeatingOn	0.9581	0.8934	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	
Converter_Setpoint	Converter_Frequency	0.9568	-0.1306	-	Setpoint and Frequency turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Not so clear (accumulation in 0 not easy to see)	
Converter_AC_Current	Converter_Frequency	0.9557	-0.1348	-	Current and Frequency turn on and off at the same time, that's No Explanation Foundcorrelation when all the data is analysed. No physical relevant information.	Logaritmic Correlation	
Converter_Power	Converter_Frequency	0.9515	-0.2348	-	Power and Frequency turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Not so clear (accumulation in 0 not easy to see)	
Converter_Power	HeatingOn	0.9435	0.1713	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	Because of Time delay between PLC signal and actual OFF of converter
Converter_WaterFlow1	Converter_WaterFlow2	0.9399	0.8977	0.9533	Same waterpump for converter water flow and coil. A change in the coil can imply a change in waterflow, and therefore in the converter waterflow too (same pump). Both waterflows inside the converter will adjust and behave in the same way.	Yes, in a specific area	
Quench1_WaterTemp	Quench2_WaterTemp	0.9007	0.9716	0.8557	The water temperature in both quenching channels behave in the same way	Yes	
Inductor_WaterFlow	JobName	0.8891	0.8934	0.8886	The job decides which coil to use, which will influecne the waterflow in it.	JobName not shown in Scatter Matrix	
Counter	HeatingCycleTime	0.8669	0.9724	0.8550	The Heating Cycle Time variable it increases with the counter if the converter is ON. Artifical variable and no physical information between these two.	Yes	
Converter_DC_Voltage	HeatingOn	0.8550	0.0332	0.0081	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	Because of Time delay between PLC signal and actual OFF of converter
Inductor_WaterTemp	BusBar_WaterTemp	0.8525	0.8759	0.8261	The water temperature in the Bus Bar and in the Coil behave in the same way. Same pump.	Yes	
HeatingCycleTime	BusBar_WaterTemp	0.7945	0.8349	0.7679	The water temperature inthe bus bar increases with the heating time.	Yes	
Converter_Frequency	Z_Axis_Velocity	0.7612	0.4954	-	The speed in the Z axis is uasually above 0 or higher while heating, except of when coming to home position after heating is finished.	Not so clear	
Converter_Setpoint	Z_Axis_Velocity	0.7597	0.3104	-	The speed in the Z axis is above 0 or higher while heating. No direct correlation.	Not so clear	
Converter_AC_Current	Z_Axis_Velocity	0.7587	0.2964	-	The speed in the Z axis is above 0 or higher while heating. No direct correlation.	Not so clear	
HeatingCycleTime	Z_Axis_Position	0.7441	0.6256	0.7801	The Z axis position increases (moves up) with the heating time. When analysing OFF, Z position is Low (home position) at the beggining) and High (top position) after turn the converter OFF.	Yes	
Z_Axis_Position	EnergyAlarm_Low	0.7430	0.2936	0.8952	Probably because Z axis position returns to its initial place and energy alarm low is activated	No	
Converter_Power	Z_Axis_Velocity	0.7354	0.0800	-	The speed in the Z axis is above 0 or higher while heating. No direct correlation.	Not so clear	
Y_Axis_Position	Inductor_WaterFlow	0.7350	0.5387	0.7480	Change of coil when using Y movement, changes waterflow in coil	Yes	
Counter	Z_Axis_Position	0.7308	0.5929	0.7455	The counter indicates the "timestamp" of the process, the axis position will increase while the counter goes up. No physical information.	Yes	

Quench1_WaterFlow	C_Axis_Warning	0.6968	0.4965	0.7973	No Explanation Found	No
Counter	BusBar_WaterTemp	0.6951	0.8080	0.6451	The water temperature increases over time. Spcially when the converter is ON	Not so clear
HeatingCycleTime	Inductor_WaterTemp	0.6688	0.8442	0.6020	The water temperature increases over time	Yes
HeatingOn	EnergyAlarm_Low	0.6525	0.2961	0.2001	The energy transferred is calculated when the converter is stopped. No physicical information	No
Converter_AC_Current	EnergyAlarm_Low	0.6428	0.3622	-	The energy transferred is calculated when the converter is stopped. No physicical information	No
Converter_Setpoint	EnergyAlarm_Low	0.6427	0.3873	-	The energy transferred is calculated when the converter is stopped. No physicical information	No
Converter_Power	EnergyAlarm_Low	0.6375	0.3922	-	The energy transferred is calculated when the converter is stopped. No physicical information	No
Converter_Frequency	EnergyAlarm_Low	0.6325	0.3673	-	The energy transferred is calculated when the converter is stopped. No physicical information	No
BusBar_WaterFlow	Quench1_WaterTemp	0.6317	0.4631	0.6805	The Busbar waterflow is influenced by a change in the coil. This can mean a different piece to hardened, with requires a higher power and therefore the quenching water temperature increases when cooling down the piece.	Not so clear
BusBar_WaterFlow	Quench2_WaterTemp	0.6299	0.4899	0.6388	The Busbar waterflow is influenced by a change in the coil. This can mean a different piece to hardened, with requires a higher power and therefore the quenching water temperature increases when cooling down the piece.	Not so clear
Z_Axis_Position	Inductor_WaterTemp	0.6264	0.5107	0.6530	The temperature in the coil behaves in the same way that the z axis position: increases when heating and moving up, and decreases when moving with no heat.	Yes
Z_Axis_Position	BusBar_WaterTemp	0.6086	0.4101	0.6240	The temperature in the bus bar behaves in the same way that the z axis position: increases when heating and moving up, and decreases when moving with no heat. Specially when cooling down.	Yes
HeatingCycleTime	Quench2_WaterFlow	0.6044	0.5553	0.6257	The quenching waterflow can be defined by differrent jobs, which can have a different heating cycle time	Not so clear
BusBar_WaterFlow	JobName	0.6031	0.3449	0.6233	The job decides which coil to use, which will influecne the waterflow in the coil, and tehrefore in the busbar.	JobName not shown in Scatter Matrix
BusBar_WaterTemp	Quench2_WaterFlow	0.5716	0.6272	0.5696	The quenching waterflow can be defined by differrent jobs, which can have different powers that influence the water temperature in the bus bar	Not so clear
Inductor_WaterTemp	C_Axis_Warning	0.5485	0.2187	0.6958	No Explanation Found	In some cases. Looks like temperature goes higher while warning is ON.
Inductor_WaterFlow	Quench2_WaterFlow	0.5448	0.6018	0.4863	The quenching waterflow can be defined by differrent jobs, which can have different coils that influecne the waterflow in the inductor	Not so clear
Counter	Inductor_WaterTemp	0.5293	0.7747	0.4479	The water temeprature in the coil increases over time while heating	Yes
HeatingCycleTime	Inductor_WaterFlow	0.5196	0.3949	0.6107	Jobs can influence the coil in use, and therefore the waterflow in the inductor. A different job can be related with a different heating cycle time.	No
HeatingCycleTime	BusBar_WaterFlow	-0.5006	-0.1528	-0.6227	Jobs can influence the coil in use, and therefore the waterflow in the inductor. The parallell water connection between inductor and busbar gives a negative relation between those two variables.	Not so clear
HeatingCycleTime	Quench2_WaterTemp	-0.5434	-0.1156	-0.6781	The heating cycle time can be job dependant. The higher the heating cycle time, it implies a higher quenching waterflow setpoint, and therefore a reduction in the quenching temperature.	Yes
Converter_WaterFlow1	Quench2_WaterTemp	-0.5813	-0.4876	-0.6048	Converter Water Flow should be almost constant. No Explanation Found	Not so clear
Converter_WaterFlow2	Quench2_WaterTemp	-0.5849	-0.4870	-0.6090	Converter Water Flow should be almost constant. No Explanation Found	Not so clear
Converter_WaterFlow1	BusBar_WaterFlow	-0.7082	-0.6666	-0.7110	Same waterpump for converter water flow and bus bar, connected in parallell. Therefore the change in one flow will have the opposite change in the other flow.	Not so clear
Converter_WaterFlow2	BusBar_WaterFlow	-0.7252	-0.6930	-0.7249	Same waterpump for converter water flow and bus bar, connected in parallell. Therefore the change in one flow will have the opposite change in the other flow.	Not so clear
Converter_DC_Voltage	Converter_Frequency	-0.7449	0.0562	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the frequency goes to 0 and the voltage to a higher stable state. Not interesting	Not so clear
Converter_DC_Voltage	Converter_AC_Current	-0.7540	-0.1203	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the current goes to 0 and the voltage to a higher stable state. Not interesting	Not so clear
Converter_Setpoint	Converter_DC_Voltage	-0.7543	-0.1210	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the setpoint goes to 0 and the voltage to a higher stable state. Not interesting	Not so clear

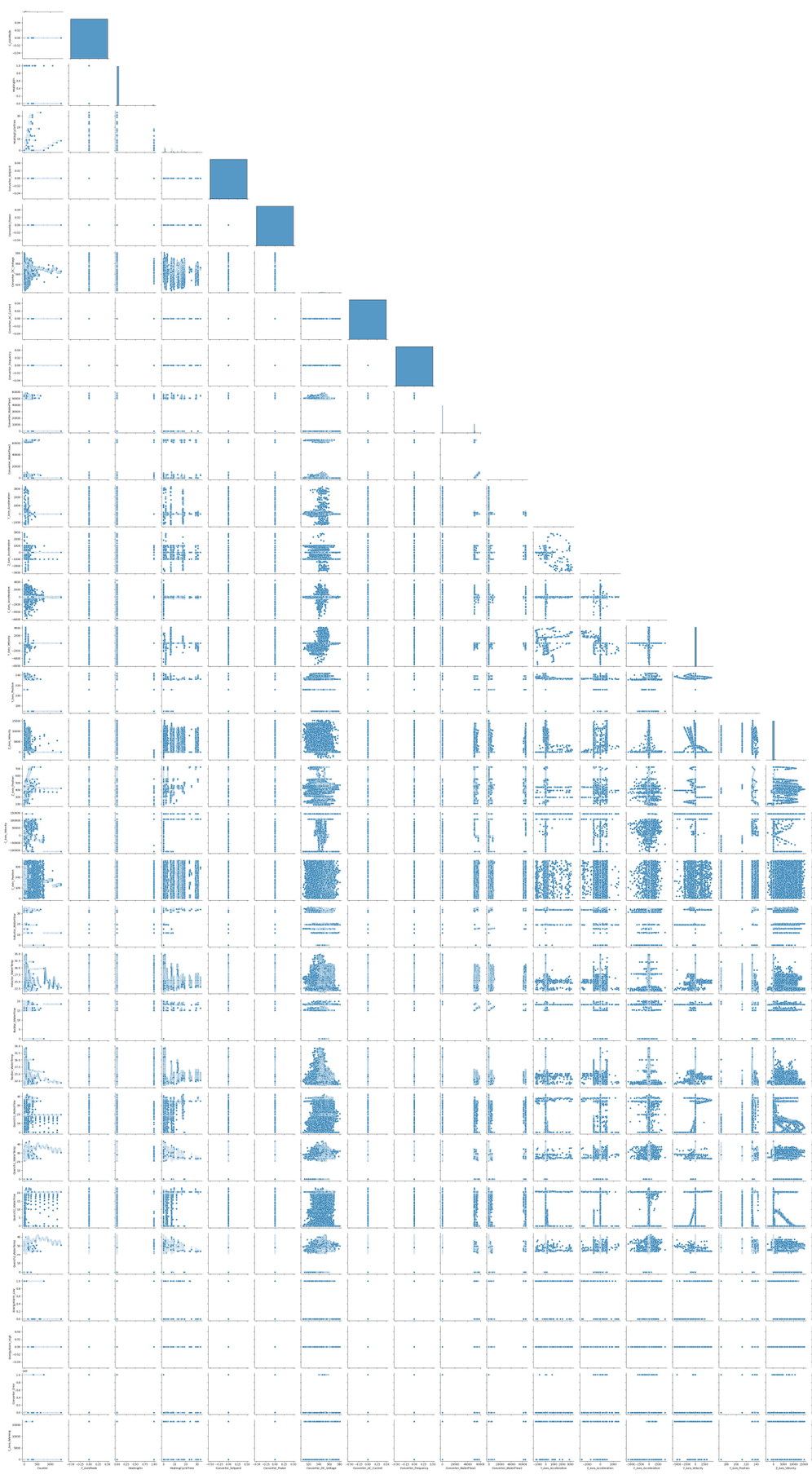
Converter_Power	Converter_DC_Voltage	-0.7600	-0.2484	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the power goes to 0 and the voltage to a higher stable state. Not interesting	Yes	Because of Time delay between PLC signal and actual OFF of converter
Y_Axis_Position	JobName	0.8567	0.8951	0.8200	The job decides if the Y axis is in use ot not and its behaviour.	JobName not shown in Scatter Matrix	
Converter_Setpoint	JobName	0.2934	0.8137	-	The configuration of the converter can be determined by the job in use	JobName not shown in Scatter Matrix	
Z_Axis_Velocity	HeatingOn	0.0946	0.7981	0.0437	The Z axe starts its movement when the heating is turned ON	Not so clear	
Quench1_WaterFlow	Quench2_WaterFlow	-0.0037	0.6910	-0.2388	Quenching water on both channels behaves in the same way	Yes	
Quench1_WaterTemp	C_Axis_Warning	0.4728	0.6470	0.4609	No physical understandable correlation	No	
Converter_AC_Current	JobName	0.2806	0.6389	-	The configuration of the converter can be determined by the job in use	JobName not shown in Scatter Matrix	
Converter_Frequency	JobName	0.3036	0.6165	-	The configuration of the converter can be determined by the job in use	JobName not shown in Scatter Matrix	
Z_Axis_Acceleration	HeatingOn	0.3805	0.6161	0.0086	The Z axe starts its movement when the heating is turned ON	No	
HeatingCycleTime	JobName	0.4932	0.2957	0.6371	The heating time can be determined by the job in use (piece to harden)	JobName not shown in Scatter Matrix	
HeatingCycleTime	Quench1_WaterTemp	-0.4853	-0.0656	-0.6258	Jobs with a higher heating cycle time, maybe result in less power applied and the water temperature in quenching does not icnrease so much.	Yes	

Scatter Matrix Client 1 - ALL





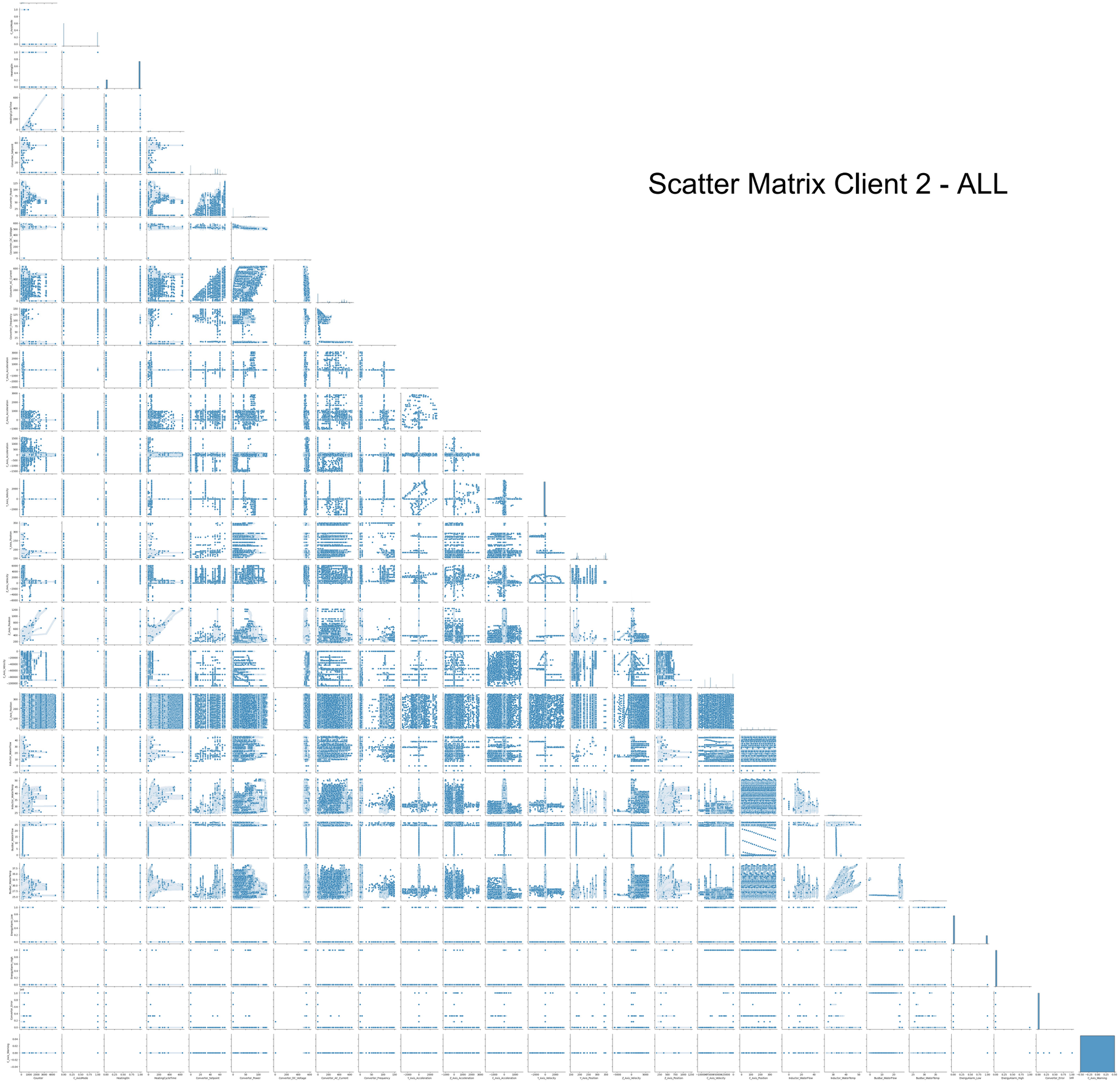
Scatter Matrix Client 1 - ON



Scatter Matrix Client 1 - OFF

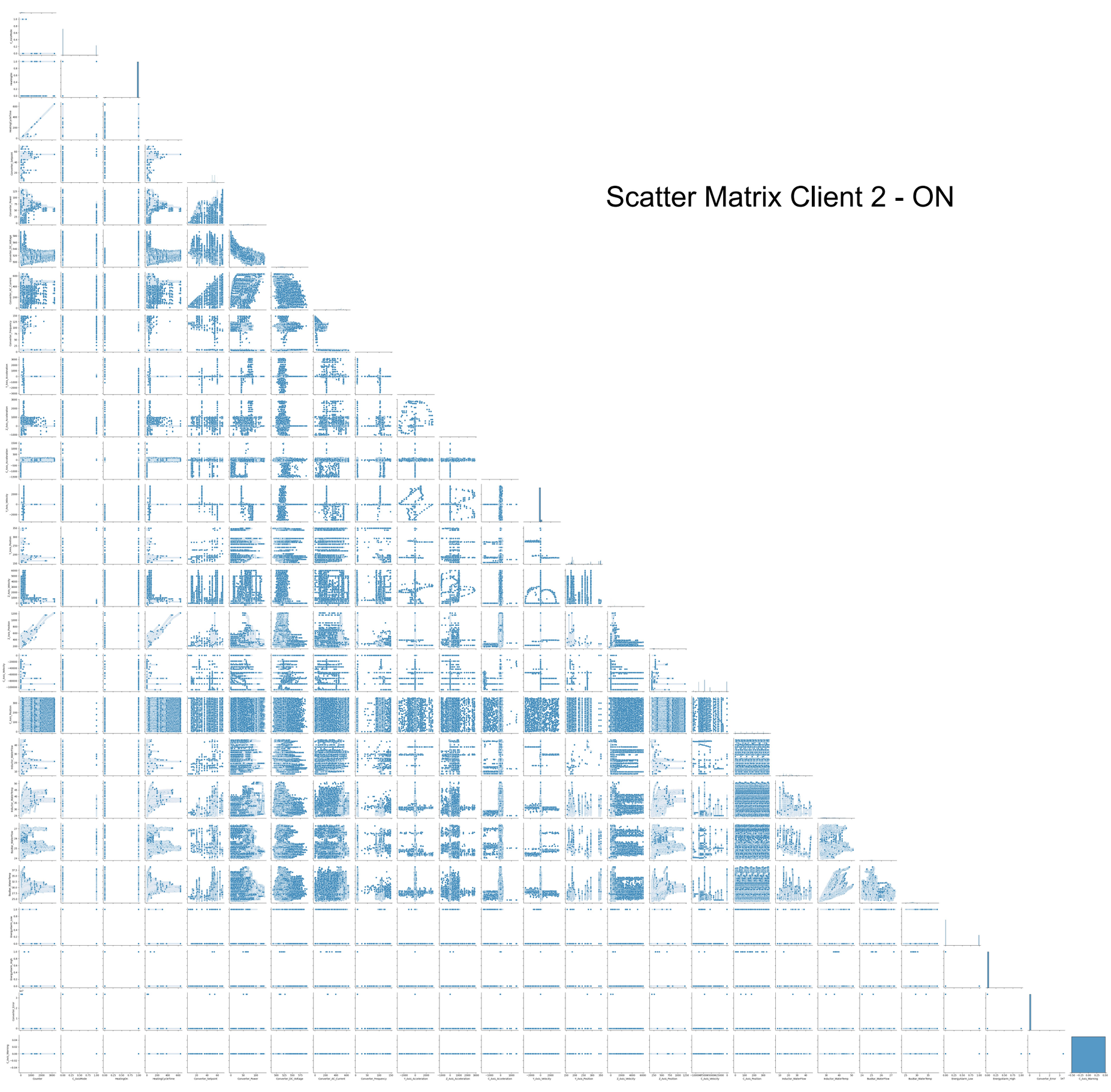
E.2 Correlations Client 2

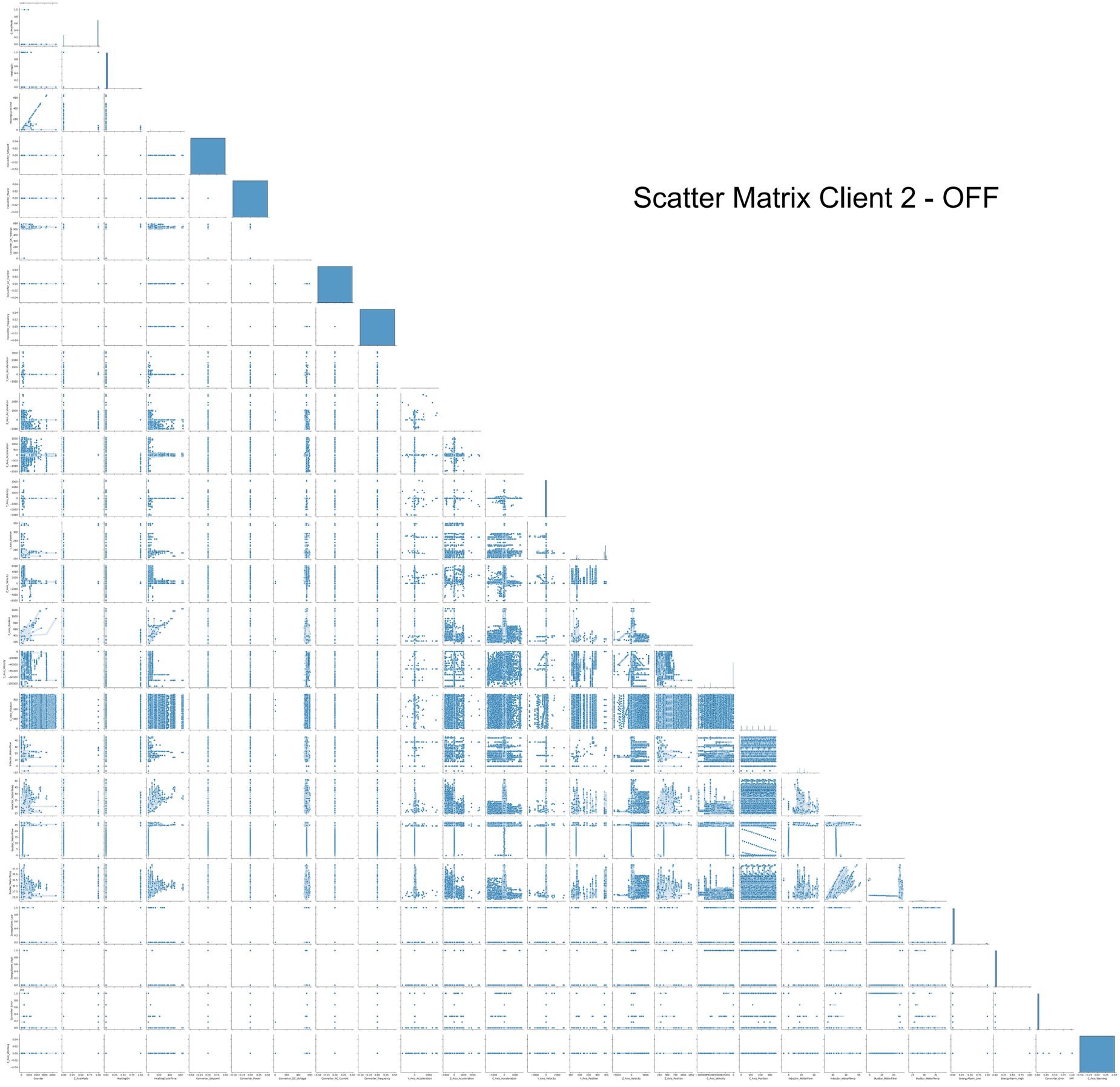
Client 2					Comments		
Variables		Correlations					
Variable 1	Variable 2	ALL	ON	OFF	Observations	Seen in Scatter Matrix	
Y_Axis_Position	C_AxisMode	1.0000	1.0000	1.0000	The Y axis has a different movement according to the C Axis Mode in use	Yes, two separate areas	
JobName	C_AxisMode	1.0000	1.0000	1.0000	The job decides which type of C Axis Mode is in use	JobName not shown in Scatter Matrix	
C_Axis_Velocity	C_AxisMode	0.9934	0.9936	0.9881	The speed of the C axis is dependant on the type of C Axis Mode in use	Yes, two separate areas	
Converter_Setpoint	HeatingOn	0.9812	0.0402	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	Because of Time delay between PLC signal and actual OFF of converter
Converter_AC_Current	HeatingOn	0.9797	0.2699	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	
Converter_Power	HeatingOn	0.9612	0.0707	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	
Converter_Setpoint	Converter_AC_Current	0.9369	0.8552	-	Setpoint and output current behave in the same way. Only present if Converter is ON	Yes	
Counter	HeatingCycleTime	0.9178	0.9989	0.4991	The Heating Cycle Time variable it increases with the counter if the converter is ON. Artifical variable and no physical information between these two.	Yes	
Y_Axis_Position	C_Axis_Velocity	0.8448	0.7943	0.7896	The Y axis movement is dependant on the C Axis Mode, and this is depdendant with the C axis velocity.	Not so clear	
Converter_Power	Converter_AC_Current	0.8306	0.6139	-	Power and current behave in the same way. Only present if Converter is ON	Not so clear	
HeatingCycleTime	Inductor_WaterTemp	0.7978	0.8405	0.5714	The water temperature increases over time	Yes	
Converter_Setpoint	Converter_Power	0.7770	0.4886	-	Setpoint and Power turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Yes	
Inductor_WaterFlow	C_AxisMode	0.7723	0.6825	0.7142	C axis Mode is job dependant, and the job decides which coil to use, which will influecne the waterflow in it.	Yes, two separate areas	
Inductor_WaterTemp	BusBar_WaterTemp	0.7628	0.7988	0.7026	The water temperature in the Bus Bar and in the Coil behave in the same way. Same pump.	Yes	
Y_Axis_Position	JobName	0.7325	0.7424	0.6986	The job decides if the Y axis is in use ot not and its behaviour.	JobName not shown in Scatter Matrix	
Converter_DC_Voltage	HeatingOn	0.7299	0.0357	0.0073	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	Because of Time delay between PLC signal and actual OFF of converter
Counter	Inductor_WaterTemp	0.6998	0.8356	0.0228	The water temeprature in the coil increases over time while heating	Yes. Growing Logaritmic?	
C_Axis_Velocity	JobName	0.6725	0.6912	0.5932	The speed of the C axis is dependant on the type of C Axis Mode in use, and therfore of the job selected.	JobName not shown in Scatter Matrix	
Inductor_WaterFlow	JobName	0.6591	0.7315	0.6210	The job decides which coil to use, which will influecne the waterflow in it.	JobName not shown in Scatter Matrix	
Counter	Z_Axis_Position	0.6581	0.6950	0.4906	The counter indicates the "timestamp" of the process, the axis position will increase while the counter goes up. No physical information.	Yes	
HeatingCycleTime	Z_Axis_Position	0.6373	0.6931	0.1202	The Z axis position increases (moves up) with the heating time.	Yes	
JobName	EnergyAlarm_Low	0.6373	0.7951	0.2732	Some jobs have more enegy alarms than others. This can be a problem with that job or that the limits/configuration of the alarms are not change along jobs.	JobName not shown in Scatter Matrix	
Y_Axis_Position	Inductor_WaterFlow	0.6158	0.5996	0.7003	Change of coil when using Y movement, changes waterflow in coil	Not so clear	
Z_Axis_Position	C_AxisMode	0.5156	0.6378	0.7056	It is possible that becuase of the usage of another type of C axis, the Z axis has to be set in a different starting position	Yes, movement range is different	
Converter_Setpoint	Converter_DC_Voltage	-0.6498	-0.2274	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the setpoint goes to 0 and the voltage to a higher stable state. Not interesting	Not so clear, but accumulation in 0	
Converter_DC_Voltage	Converter_AC_Current	-0.6645	-0.2629	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the current goes to 0 and the voltage to a higher stable state. Not interesting	Not so clear, but accumulation in 0	
Converter_Power	Converter_DC_Voltage	-0.7469	-0.4500	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the power goes to 0 and the voltage to a higher stable state. Not interesting	Yes and accumulation in 0. / Logaritmic correlation when ON.	
Converter_Setpoint	Y_Axis_Position	0.1380	0.6416	-	A higher setpoint is used when the Y axis moves in a higher range. Not very relevant, a matter of configuration.	Not so clear	
Converter_Power	Converter_Frequency	0.2675	-0.7194	-	SMH machine. Higher frequency range, has a lower maximum power output than lower frequency range, which has a higher maximum power output	Not so clear	



Scatter Matrix Client 2 - ALL

Scatter Matrix Client 2 - ON



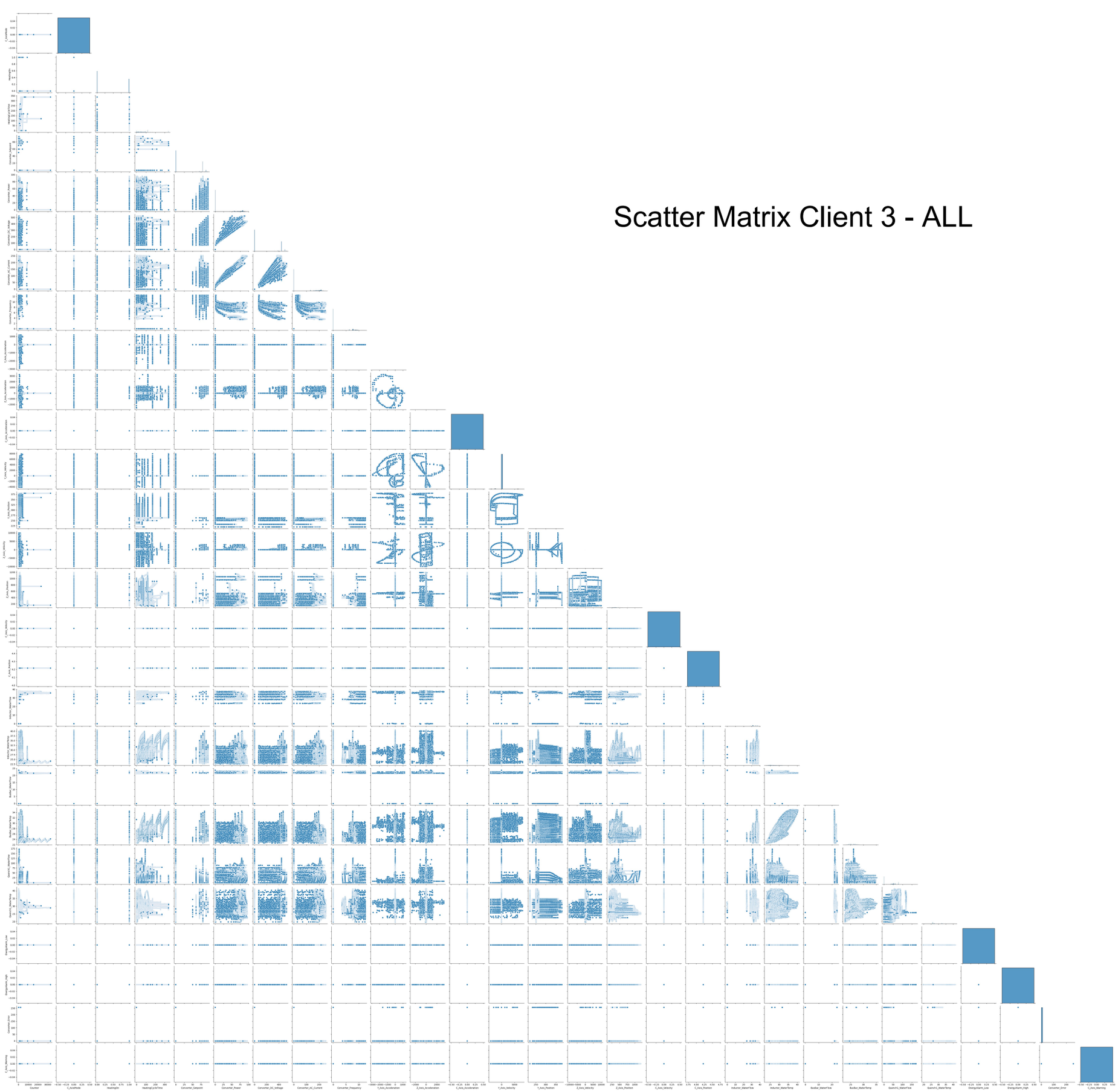


Scatter Matrix Client 2 - OFF

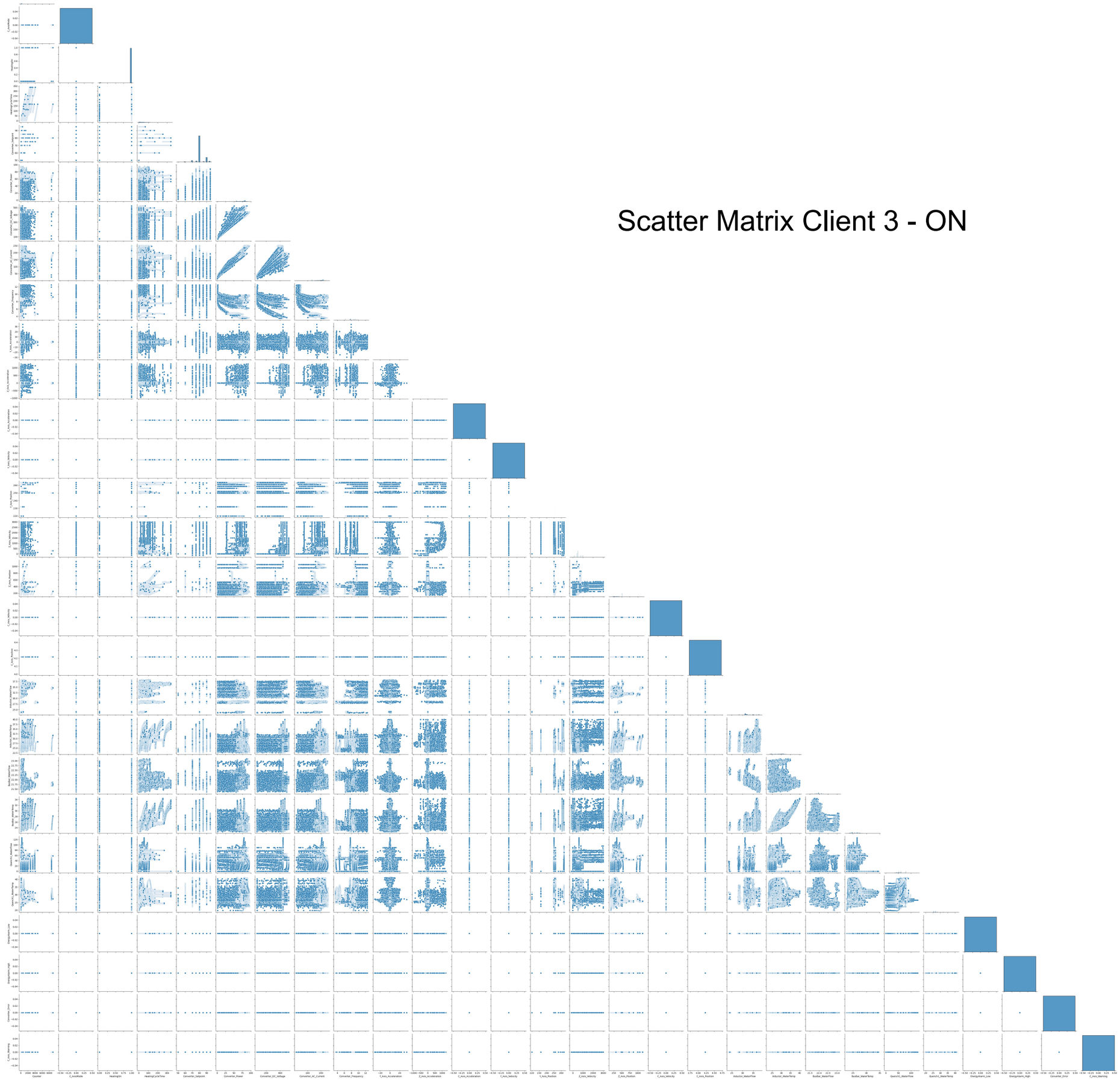
E.3 Correlations Client 3

Client 3					Comments		
Variables		Correlations					
Variable 1	Variable 2	ALL	ON	OFF	Observations	Seen in Scatter Matrix	
Inductor_WaterFlow	Converter_Error	0.9999997	-	0.9999995	Maybe converter error related to coil in use (which influences inductor waterflow)?	Yes, disrupted flow when error	
Converter_Frequency	HeatingOn	0.9972	0.8648	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	Because of Time delay between PLC signal and actual OFF of converter
Converter_DC_Voltage	HeatingOn	0.9963	0.7710	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	
Converter_AC_Current	HeatingOn	0.9959	0.7588	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	
Converter_Power	Converter_AC_Current	0.9953	0.9398	-	Power and current behave in the same way.	Strong Linear Correlation (also when ON)	
Converter_Power	HeatingOn	0.9945	0.7461	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	Because of Time delay between PLC signal and actual OFF of converter
Converter_Setpoint	HeatingOn	0.9906	0.1083	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No	
Converter_Setpoint	Converter_DC_Voltage	0.9812	0.7208	-	Normally setpoint is a current or power setpoint, but in the parallel converter is the voltage that is controlled. Therefore directed related with voltage	Yes	
Converter_AC_Current	Converter_Frequency	0.9341	0.1492	-	Current and Frequency turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Strong Logaritmic Correlation (also when ON)	
Converter_Setpoint	Converter_Power	0.9331	-0.0767	-	Setpoint and Power turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Not so clear	
Converter_Power	Converter_Frequency	0.9326	0.1300	-	Power and Frequency turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Strong Logaritmic Correlation (also when ON)	
Converter_Power	Converter_DC_Voltage	0.9257	0.0376	-	Voltage is controlled to obtain a certain power, therefore voltage changes in the same direction as the power (Parallell)	Strong Linear Correlation (also when ON)	
BusBar_WaterFlow	Converter_Error	0.9209	-	0.9208	Maybe converter error related to coil in use (which influences bus bar waterflow)?	Yes, disrupted flow when error	
Converter_Setpoint	Converter_AC_Current	0.9192	-0.3074	-	Setpoint and AC Current turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Not so clear	
Converter_Setpoint	Converter_Frequency	0.9176	-0.3349	-	Setpoint and Frequency turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information.	Not so clear	
Converter_DC_Voltage	Converter_AC_Current	0.9114	-0.1468	-	Voltage and Current behave in the same way in Parallell converter. No Explanation Found that it does not happen when ON.	Strong Linear Correlation (also when ON)	
Converter_DC_Voltage	Converter_Frequency	0.9030	-0.2565	-	Voltage turns ON with current and converter starts from a high frequency to match the desired resonance frequency. No Explanation Found that it does not happen when ON.	Logaritmic Correlation (also when ON)	
Counter	HeatingCycleTime	0.8664	0.9279	0.8533	The Heating Cycle Time variable it increases with the counter if the converter is ON. Artifical variable and no physical information between these two.	Yes	
Inductor_WaterFlow	JobName	0.8440	0.8304	0.8413	The job decides which coil to use, which will influecne the waterflow in it.	JobName not in Scatter Matrix	
Inductor_WaterTemp	BusBar_WaterTemp	0.8149	0.8861	0.8846	The water temperature in the Bus Bar and in the Coil behave in the same way. Same pump.	Yes	
Z_Axis_Velocity	HeatingOn	0.6565	0.2242	-	The Z axe starts its movement when the heating is turned ON	Reduced movement range when heating ON	
Converter_AC_Current	Z_Axis_Velocity	0.6178	0.5411	-	The speed in the Z axis is above 0 or higher while heating. No direct correlation.	No	
Converter_Power	Z_Axis_Velocity	0.6164	0.5496	-	The speed in the Z axis is above 0 or higher while heating. No direct correlation.	No	
Inductor_WaterFlow	BusBar_WaterFlow	-0.7952	-0.7476	-0.8172	The flow in this two areas change in the same way if the coil is changed	Yes	
HeatingCycleTime	JobName	0.4475	0.3234	0.6330	The job selected will determine a higher or smaller heating cycle time	JobName not in Scatter Matrix	
Z_Axis_Position	JobName	0.4825	0.5369	0.6168	The job selected will determine the behaviour of the Z axe	JobName not in Scatter Matrix	
Y_Axis_Position	JobName	0.4748	0.8552	0.5121	The job decides if the Y axis is in use ot not and its behaviour.	JobName not in Scatter Matrix	
HeatingCycleTime	Inductor_WaterTemp	0.1598	0.8167	-0.0723	The water temperature increases over time	Yes	
Converter_Setpoint	JobName	0.4892	0.8044	-	The job selected will determine a higher or smaller setpoint in the converter	JobName not in Scatter Matrix	
HeatingCycleTime	BusBar_WaterTemp	0.3574	0.7926	0.0962	The water temperature inthe bus bar increases with the heating time.	Yes	
Counter	BusBar_WaterTemp	0.1646	0.7431	-0.1919	The water temperature increases over time.	Yes	
Counter	Inductor_WaterTemp	-0.1160	0.7313	-0.4009	The water temeprature in the coil increases over time while heating	Yes	
Converter_Frequency	JobName	0.4001	0.7155	-	The job can determine which coil to use, and this will influence the frequency of the converter	JobName not in Scatter Matrix	
Counter	Z_Axis_Position	0.4729	0.7065	0.2855	The counter indicates the "timestamp" of the process, the axis position will increase while the counter goes up. No physical information.	Yes	

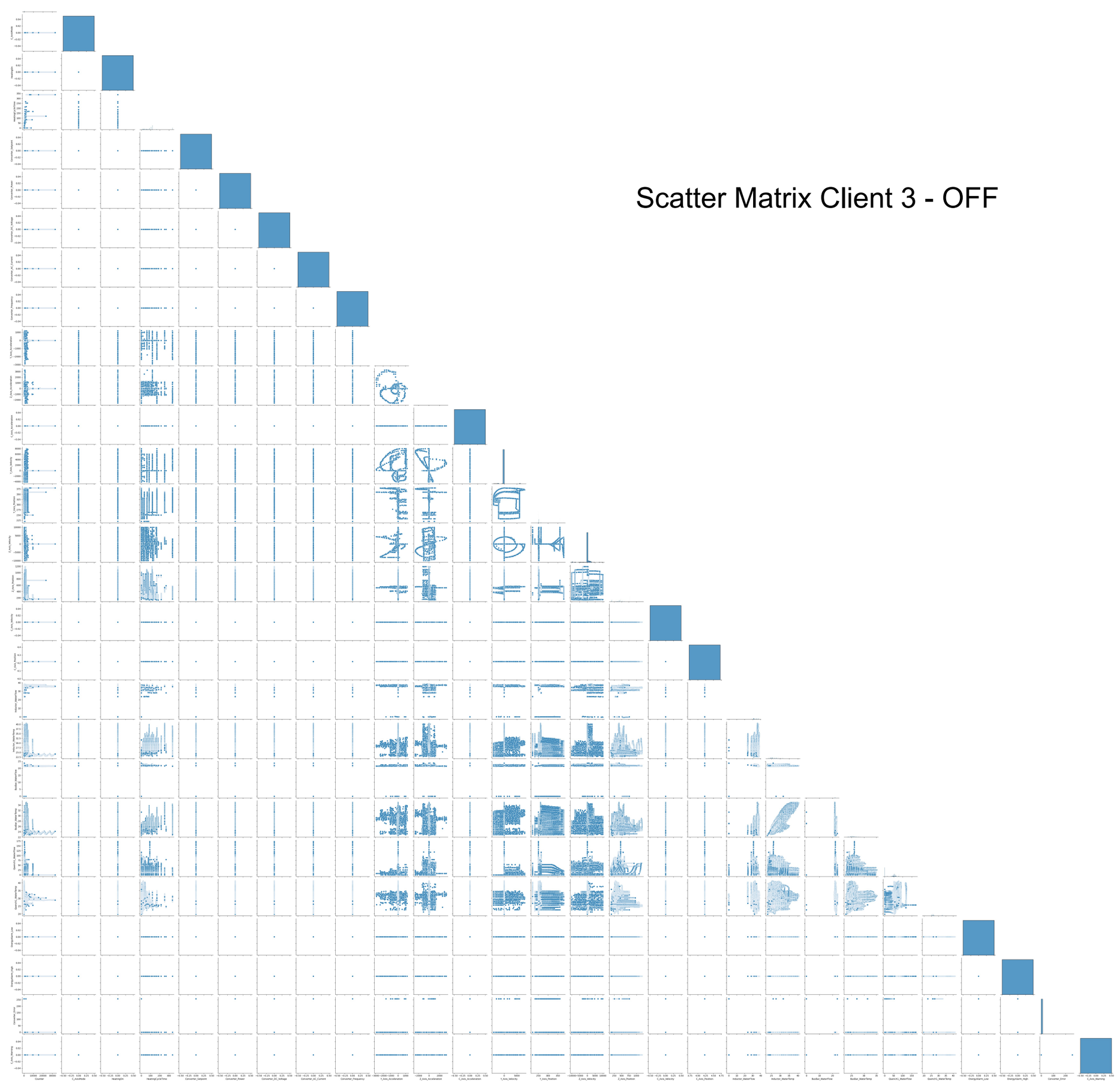
Scatter Matrix Client 3 - ALL



Scatter Matrix Client 3 - ON



Scatter Matrix Client 3 - OFF



E.4 Common Correlations

Variables		Correlations									Comments	
		Client 1			Client 2			Client 3				
Variable 1	Variable 2	ALL	ON	OFF	ALL	ON	OFF	ALL	ON	OFF	Observations	Seen in Scatter Matrix
Converter_Setpoint	Converter_AC_Current	0.9988	0.9684	-	0.9369	0.8552	-	0.9192	-0.3074	-	- Setpoint and output current behave in the same way (serial). - Setpoint and AC Current turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information. (parallel)	Yes / Not so clear (parallel)
Converter_Setpoint	Converter_Power	0.9823	0.5386	-	0.7770	0.4886	-	0.9331	-0.0767	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the setpoint goes to 0 and so does the power. Not interesting	Yes / Not so clear (parallel)
Converter_Power	Converter_AC_Current	0.9817	0.5321	-	0.8306	0.6139	-	0.9953	0.9398	-	Power and current behave in the same way.	Not so clear / Strong linear correlation (parallel)
Converter_AC_Current	HeatingOn	0.9592	0.2610	-	0.9797	0.2699	-	0.9959	0.7588	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Converter_Setpoint	HeatingOn	0.9581	0.1459	-	0.9812	0.0402	-	0.9906	0.1083	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Converter_Power	Converter_Frequency	0.9515	-0.2348	-	0.2675	-0.7194	-	0.9326	0.1300	-	- Power and Frequency turn on and off at the same time, that's why correlation when all the data is analysed. No physical relevant information. (Client 1 and 3) - For bigger pieces that can requiere a higher power from the converter, the coil needs to be changed to a bigger one (to fit the piece) and this will result in a smaller resonance frequency.(Client 2)	Not so clear / Strong logaritmic correlation (parallel)
Converter_Power	HeatingOn	0.9435	0.1713	-	0.9612	0.0707	-	0.9945	0.7461	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Counter	HeatingCycleTime	0.8669	0.9724	0.8550	0.9178	0.9989	0.4991	0.8664	0.9279	0.8533	The Heating Cycle Time variable it increases with the counter if the converter is ON. Artifical variable and no physical information between these two.	Yes
Converter_DC_Voltage	HeatingOn	0.8550	0.0332	0.0081	0.7299	0.0357	0.0073	0.9963	0.7710	-	Signal from PLC activates converter with time delay, no relevant physical information. The correlations is only in all because is when the change is visible.	No
Inductor_WaterTemp	BusBar_WaterTemp	0.8525	0.8759	0.8261	0.7628	0.7988	0.7026	0.8149	0.8861	0.8846	The water temperature in the Bus Bar and in the Coil behave in the same way. Same pump.	Yes
Counter	Z_Axis_Position	0.7308	0.5929	0.7455	0.6581	0.6950	0.4906	0.4729	0.7065	0.2855	The counter indicates the "timestamp" of the process, the axis position will increase while the counter goes up. No physical information.	Yes
HeatingCycleTime	Inductor_WaterTemp	0.6688	0.8442	0.6020	0.7978	0.8405	0.5714	0.1598	0.8167	-0.0723	The water temeprature in the coil increases over time while heating is ON	Yes
Counter	Inductor_WaterTemp	0.5293	0.7747	0.4479	0.6998	0.8356	0.0228	-0.1160	0.7313	-0.4009	The water temeprature in the coil increases over tthe process (specially when converter is ON)	Yes
Converter_DC_Voltage	Converter_AC_Current	-0.7540	-0.1203	-	-0.6645	-0.2629	-	0.9114	-0.1468	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the current goes to 0 and the voltage to a higher stable state (serial) or to OFF (parallel). Not interesting	Not so clear / Strong linear correlation also when ON (parallel)
Converter_Setpoint	Converter_DC_Voltage	-0.7543	-0.1210	-	-0.6498	-0.2274	-	0.9812	0.7208	-	- This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the setpoint goes to 0 and the voltage to a higher stable state. Not interesting (serial) - Normally setpoint is a current or power setpoint, but in the parallel converter is the voltage that is controlled. Therefore directed related with voltage (parallel)	No / Yes (parallel)
Converter_Power	Converter_DC_Voltage	-0.7600	-0.2484	-	-0.7469	-0.4500	-	0.9257	0.0376	-	This correlations would have been interesting if it was while the converter was ON. When analysing all data it just indicates that when the converter turns OFF, the power goes to 0 and the voltage to a higher stable state (serial) or to OFF (parallel). Not interesting	Yes / Strong lienar correlation also when ON (parallel)