

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Optimization of the new alarm system implemented on PowerStudio SCADA

Raquel Medeiros da Ponte

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Rui Pedro Ferreira Pinto

July 28, 2023

Resumo

Dada a atual situação geopolítica e a atual crise energética global, há uma necessidade cada vez mais premente de reduzir o consumo de energia, especialmente na indústria. Devido à produção contínua e em massa em que operam os fornos e as linhas de produção, a indústria de fabrico de embalagens de vidro foi particularmente afetada pelo aumento dos preços da energia no último ano, sendo provável que esta tendência se mantenha.

Para gerir os gastos energéticos da sua infraestrutura, a BA Glass Avintes tomou medidas proativas, substituindo o sistema de Controlo de Supervisão e Aquisição de Dados (SCADA) existente, baseado na tecnologia *Siemens S7*, pelo *software* de gestão energética *PowerStudio SCADA* da *Circutor*. Este *software* oferece estudos energéticos abrangentes, análise de rácios de produção (consumo de energia por unidade produzida), gestão da qualidade da energia, geração automática de relatórios e capacidades de visualização de dados.

Foram identificados dois problemas principais: a subutilização do sistema SCADA do *PowerStudio*, particularmente em termos da sua funcionalidade de alarmística, e a ausência de ferramentas para prever com exatidão o consumo dos fornos. Uma vez que os fornos constituem a principal fonte de consumo de energia na fábrica, obter uma compreensão abrangente do seu consumo de energia é um passo crucial para permitir a tomada de decisões informadas.

O primeiro objetivo desta dissertação foi aproveitar todo o potencial do *PowerStudio SCADA* para desenvolver um sistema eficiente e de fácil utilização. O primeiro passo consistiu em determinar a arquitetura do sistema *PowerStudio SCADA*. Um aspeto crucial foi a definição de um protocolo para integrar os sistemas antigos existentes com o *PowerStudio*, permitindo a unificação da arquitetura da maquinaria da fábrica. O *software* do *PowerStudio* utiliza o protocolo OPC (*Open Platform Communications*), uma vez que é um método seguro, fiável e aberto para a troca de informações entre clientes e servidores, incluindo dispositivos industriais como controladores lógicos programáveis (PLC) e sensores e atuadores (S&A).

Para estabelecer uma comunicação perfeita entre vários equipamentos no chão de fábrica e o sistema SCADA, foram utilizados contadores inteligentes e tecnologia da Internet das Coisas (IoT). A conectividade *Ethernet* foi empregue para garantir a evolução da fábrica de Avintes em linha com os princípios da Indústria 4.0 (I4.0). Esta integração de dispositivos e recolha de dados constituiu a base para a recolha dos dados que seriam apresentados no novo e melhorado sistema SCADA. Para garantir uma recolha de dados sem falhas, foram implementadas melhorias no sistema de alarmes do SCADA para detetar eventuais anomalias nos contadores e melhorar a monitorização da energia.

O segundo objetivo desta investigação era utilizar os dados adquiridos com o *PowerStudio SCADA* e as ferramentas de aprendizagem computacional para realizar uma análise abrangente. Esta análise foi crucial para identificar valores anómalos e garantir um conjunto de dados padronizado e consistente para o desenvolvimento de uma ferramenta de aprendizagem supervisionada. O objetivo desta ferramenta era prever com precisão o consumo de gás e eletricidade para os três fornos da fábrica de Avintes. Durante esta fase, surgiram duas questões importantes: Em primeiro lugar,

qual o modelo de regressão que produzia os resultados mais exatos, especificamente adaptados ao contexto da fábrica de Avintes? Em segundo lugar, qual seria a combinação ótima de períodos de tempo de treino e de previsão, que resultaria no resultado mais preciso e exato?

Este estudo comparou diferentes modelos de regressão para prever o consumo do forno na indústria de fabrico de vidro. As conclusões sugerem que os modelos de regressão linear podem não ser adequados para esta aplicação, enquanto os modelos baseados em árvores apresentam resultados promissores, em particular o modelo de árvore de decisão.

O modelo de árvore de decisão tem melhor desempenho para períodos de previsão mais longos, capturando efetivamente padrões de dados complexos. Por outro lado, a regressão polinomial, quando combinada com o *GridSearchCV*, produz melhores resultados para períodos de previsão mais curtos.

Em conclusão, este estudo salienta a importância de um conjunto de dados de treino mais alargado e de um período de previsão mais curto para melhorar o desempenho do modelo. Com um período de previsão de duas semanas e dados de treino de três meses, o modelo de árvore de decisão é selecionado como a ferramenta de previsão final devido à sua eficiência computacional superior e precisão satisfatória.

Palavras-chave Embalagem de vidro, SCADA, OPC, Aprendizagem Computacional, Gestão Energética Baseada em Dados, Monitorização Energética em SCADA, Análise de Alarmística SCADA

Abstract

Given the current geopolitical situation and the current global energy crisis, there is an ever-more pressing need to reduce energy consumption, especially in industry. Due to the continuous and mass production that furnaces and production lines operate in, the glass packaging manufacturing industry has been particularly impacted by the rise of energy prices in the last year, and the likelihood that this trend will continue.

To manage the energy expenditure within its infrastructure, BA Glass Avintes has taken proactive measures by replacing its existing Supervisory Control and Data Acquisition (SCADA) system based on *Siemens S7* technology with *Circuitor's PowerStudio SCADA* energy management software. This software offers comprehensive energy studies, production ratios analysis (energy consumption per unit produced), power quality management, automatic report generation, and data visualization capabilities.

Two primary issues were identified: the underutilization of *PowerStudio's* SCADA system, particularly in terms of its alarm functionality, and the absence of tools for accurately predicting furnace consumption. As the furnaces constitute the primary energy expenditure source in the plant, gaining a comprehensive understanding of their energy consumption is a crucial step toward enabling informed decision-making.

The first goal of this dissertation was to leverage the full potential of *PowerStudio SCADA* to develop an efficient and user-friendly system. The first step involved determining the architecture of the *PowerStudio SCADA* system. A crucial aspect was defining a protocol to integrate the existing legacy systems with *PowerStudio*, enabling the unification of the factory machinery architecture. *PowerStudio's* software utilizes Open Platform Communications (OPC) protocol since it's a secure, reliable, and open method for information exchange between clients and servers, including industrial devices such as programmable logic controllers (PLC) and sensors and actuators (S&A).

To establish seamless communication between various equipment on the plant floor and the SCADA system, smart meters and Internet of Things (IoT) technology were utilized. Ethernet connectivity was employed to ensure the Avintes plant's evolution in line with the principles of Industry 4.0 (I4.0). This integration of devices and data collection formed the foundation for collecting the data, which would be presented in the new and improved SCADA system. To guarantee flawless data collection, the SCADA's alarm system enhancements were implemented to detect any potential malfunctions in the flow meters and improve energy monitoring.

The second objective of this research aimed to utilize data acquired from *PowerStudio SCADA* and machine learning tools to perform a comprehensive analysis. This analysis was crucial in identifying abnormal values and ensuring a standardized and consistent dataset for developing a supervised learning tool. This tool's purpose was to accurately predict gas and electricity consumption for the three furnaces at the Avintes plant. During this phase, two significant questions emerged: Firstly, which regression model yielded the most accurate results specifically tailored

to the Avintes plant context? Secondly, what would be the optimal combination of training and prediction time periods, resulting in the most precise and accurate output?

This study compared different regression models for predicting furnace consumption in the glass manufacturing industry. The findings suggest that linear regression models may not be suitable for this application, while tree-based models show promising results, particularly the decision tree model.

The decision tree model performs better for longer prediction periods, effectively capturing complex data patterns. On the other hand, polynomial regression, when combined with *Grid-SearchCV*, yields better results for shorter prediction times.

In conclusion, the study highlights the importance of a larger training dataset and a shorter prediction period for improved model performance. With a two-week prediction period and three-month training data, the decision tree model is selected as the final prediction tool due to its superior computational efficiency and satisfactory accuracy.

Keywords Glass Packaging, SCADA, OPC, Machine Learning, Data-Driven Energy Management, SCADA Energy Monitoring, SCADA Alarm Analysis

Agradecimentos

Gostaria de expressar a minha profunda gratidão e apreço a todos os que contribuíram para a conclusão bem sucedida da minha dissertação e para o culminar do meu percurso de cinco anos em Engenharia Eletrotécnica e de Computadores. Este marco académico não teria sido possível sem o apoio, a orientação e o encorajamento de muitas pessoas.

Em primeiro lugar e acima de tudo, os meus sinceros agradecimentos ao meu orientador, Prof. Rui Pinto. A sua experiência, dedicação e *feedback* perspicaz ao longo deste trabalho de investigação foram inestimáveis.

Estou profundamente grata à BA Glass, especialmente ao Departamento de Manutenção Eléctrica e Instrumentação e à Equipa de *Data Science*. O seu empenho na transmissão de conhecimentos e na promoção de um ambiente de aprendizagem estimulante foi fundamental para o meu desenvolvimento como engenheira. Agradeço aos engenheiros Tiago Meireles, João Alves e Paulo Gomes pelo seu apoio, experiência e disponibilidade.

Gostaria de agradecer o contributo inestimável dos meus colegas e amigos que me apoiaram ao longo deste árduo percurso. A vossa amizade, camaradagem e discussões intelectuais motivaram-me e inspiraram-me. As nossas inúmeras sessões de estudo noturnas, sessões de *Discord*, churrascos e conversas foram essenciais para o meu crescimento pessoal e académico. Como prometido, uma palavra de apreço especial ao João Carlos Pimentel, à Gabriella Fernandes e ao Marco Belim pela vossa ajuda nos relatórios, projectos e cadeiras mais desafiantes.

Obrigada aos meus colegas de casa e amigos. As refeições partilhadas e os momentos de riso que partilhámos criaram uma sensação de lar e de camaradagem que vou guardar para sempre.

Quero também expressar o meu mais profundo agradecimento ao João Monteiro. O seu encorajamento e confiança em mim têm sido uma fonte constante de força e motivação. A sua paciência, compreensão e apoio durante as inúmeras horas que dediquei a esta tese foram notáveis. A tua presença ao meu lado, celebrando cada marco, tornou esta viagem ainda mais significativa.

Por último, a minha sincera gratidão estende-se à minha família pelo seu apoio e compreensão inabaláveis. Estou grata pelos sacrifícios que fizeram e pelo encorajamento sem fim que me deram, mesmo nos momentos mais difíceis. Este feito não teria sido possível sem vocês.

A todos os mencionados e aos inúmeros outros que me apoiaram de formas grandes e pequenas, os meus sinceros agradecimentos. A vossa orientação, encorajamento e crença nas minhas capacidades desempenharam um papel indispensável na formação da engenheira que sou hoje.

Ao embarcar no próximo capítulo da minha carreira, levo comigo as lições aprendidas, os conhecimentos adquiridos e as relações estabelecidas durante estes últimos cinco anos. É com profunda gratidão e humildade que reconheço o esforço colectivo que levou à conclusão desta dissertação e à realização do meu sonho de me tornar engenheira eletrotécnica.

Obrigada a todos.

Raquel Ponte

*“End?
No, the journey doesn’t end here.”*

J.R.R. Tolkien

Contents

1	Introduction	1
1.1	Context	1
1.1.1	BA Glass History	1
1.1.2	Glass Manufacturing Process	3
1.1.3	Energy Sustainability in Industry 5.0	4
1.2	Motivation	5
1.3	Problem Definition	6
1.4	Objectives	7
1.5	Dissertation Structure	7
2	State of the Art	9
2.1	Supervisory Control And Data Acquisition System and Energy Monitoring	9
2.1.1	Digital Twin (DT)	10
2.1.2	DINASORE	11
2.1.3	<i>PowerStudio SCADA</i>	12
2.1.4	OPC	13
2.2	Data-Driven Energy Analysis	14
2.2.1	Regression-Based Supervised Methods	15
2.2.2	Data Science Programming Languages	18
2.2.3	Hyperparameter Optimization	20
2.2.4	Transfer Learning	21
3	<i>PowerStudio</i> Data Collection and Alarmistics	23
3.1	Integration of <i>PowerStudio SCADA</i> and OPC Router’s Modular System	23
3.1.1	Smart energy meters and IoT	24
3.1.2	Graphic User Interface	25
3.1.3	External Data Analytics Module (OPC Router)	26
3.2	Integration of Alarmistics and SCADA	27
4	Data-driven Energy Management, Monitoring and Forecasting	29
4.1	Dataset Analysis	29
4.2	Analysis and Forecasting for Energy Consumption Prediction using Data Log Values	32
4.2.1	Impact Analysis of Features on Furnace Consumption	33
4.2.2	Development of Predictive Models for Furnace Consumption	36
4.3	Exploring an Approach for Furnace Consumption Optimization	36
4.4	Estimating Consumption using Models Transferred from Other Furnaces	40

5	Validation and Evaluation	41
5.1	<i>PowerStudio</i> Validation Methodology	41
5.2	Furnace Consumption Prediction Tool Testing Methodology	42
5.2.1	First testing scenario	43
5.2.2	Second testing scenario	45
5.3	Results and evaluation	47
6	Conclusions and Future Work	49
6.1	Discussion	49
6.1.1	Work Contributions	51
6.1.2	Limitations	52
6.2	Future Work	53
A	Images	55
A.1	GUI Improvements (3.1.2)	56
A.2	Collected Data vs. Filtered Data	61
A.3	Comparison of Regression Models for Predicting Specific Consumption based on Pull	62
A.4	Transfer Learning: Comparison of Model Performance with Data Collected from a Different Furnace	63
A.5	First Test Scenario Results	65
B	Tables	67
C	Code	75
C.1	C# Script for Detection of Flow Meter Malfunctions (3.2)	75
C.2	Final Application (6.1)	76
	References	81

List of Figures

1.1	BA Glass’s plants location	2
1.2	Glass container manufacturing process	4
1.3	Evolution of gas prices for non-household consumers (source: EUROSTAT, 2023 [7])	5
1.4	Evolution of electricity prices for non-household consumers (source: EUROSTAT, 2023 [7])	6
1.5	Workflow with document structure	8
2.1	CPSs and DTs in manufacturing (source: Fei Tao <i>et al.</i> , 2019 [17])	11
2.2	Illustration of <i>PowerStudio SCADA</i> OPC and Modbus communications (source: Circutor, 2016 [24])	12
2.3	Illustration of OPC Client-Server communication (source: Instrumentation Tools, 2023 [27])	14
2.4	Achieving a well-balanced model	17
2.5	Grid search workflow (source: Pedregosa <i>et al.</i> , 2011 [37])	21
3.1	SCADA devices communication scheme	25
3.2	Alarm setpoint definition screen	26
3.3	Connection created in OPC Router to register SCADA’s values in Excel	27
3.4	Connection in OPC Router to trigger alarms when a flow meter malfunction is detected	28
4.1	Comparison between actual pull and planned pull for each furnace	31
4.2	Feature correlations	34
4.3	Total (gas and electrical) specific consumption of Avinte’s furnaces by color	35
4.4	Boosting percentage vs. pull in tonnes	38
4.5	Analysis of boosting with cullet within a 5% range	39
6.1	Graph result from running AV4’s energy consumption prediction tool	51
A.1	Automatic gas report	56
A.2	Main screen	57
A.3	Plant’s gas consumption screen	58
A.4	Production lines’ gas consumption screen	59
A.5	Furnace AV2 cooling system screen	60
A.6	Comparison of collected and filtered total consumption, pull and specific consumption data (4.1)	61
A.7	Relationship between pull and specific consumption: regression performance comparison (4.3)	62

A.8 Comparison of performance of one furnace's model with another furnace's data for a one-year interval (4.4)	63
A.9 Comparison of performance of one furnace's model with another furnace's data for a one-month interval (4.4)	64
A.10 Regression models performance (5.3)	65

List of Tables

4.1	Percentage of filtered data for each furnace	32
4.2	Average of total specific consumption of each furnace by color	35
4.3	Comparison of Excel's calculated regression and <i>sklearn</i> 's linear regression for analyzing gas and boosting in furnaces	37
4.4	Comparison of performance metrics for AV2's model trained with AV2 and AV4 data, and vice-versa	40
5.1	Gas flowmeters values registration table	42
B.1	Section of "Registos consumos_auto_1" Excel file (4.1)	68
B.2	Section of the Excel file containing the log of pull planned for the following week (4.1)	69
B.3	Metrics obtained for various regression models for each furnace (5.3)	70
B.4	AV2's results obtained from the second test scenario (5.3)	71
B.5	AV4's results obtained from the second test scenario (5.3)	72
B.6	AV5's results obtained from the second test scenario (5.3)	73
B.7	Results from the final application (6.1)	74

Abbreviations

AE	Alarms & Events
AI	Artificial Intelligence
API	Application Programming Interface
CPPS	Cyber-Physical Production Systems
DA	Data Access
DCOM	Distributed Component Object Model
GUI	Graphic User Interface
GSM	Global System for Mobile Communications
HDA	Historical Data Access
HMI	Human Machine Interface
I4.0	Industry 4.0
I5.0	Industry 5.0
IoT	Internet of Things
IP	Internet Protocol
IS	Individual Section
LASSO	Least Absolute Shrinkage and Selection Operator
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Square Error
OPC	Open Platform Communications
PLC	Programmable Logic Controller
PSS	PowerStudio SCADA
RMSE	Root Mean Square Error
S&A	Sensors and Actuators
SAV	Sum of Absolute Values
SCADA	Supervisory Control And Data Acquisition
SD	Standard Deviation
SOA	Service Oriented Architecture
SS	Sum of Squares
TCP	Transmission Control Protocol
UA	Unified Architecture
r^2	Coefficient of determination

Chapter 1

Introduction

This study is part of a curricular dissertation within the engineering area of industrial automation. The goal is to continue migrating the old SCADA system in BA Glass to the new *PowerStudio SCADA* (PSS) one and incorporate all the new systems and features installed on the plant floor.

We are currently on the brink of Industry 4.0 (I4.0), the fourth industrial revolution in which manufacturing and other industrial processes are conceived as “smart environments” where machines, sensors, and actuators are interconnected to enable collaboration, monitoring, and control [1]. SCADA systems are essential to I4.0 as they allow real-time monitoring, control, and optimization of industrial processes.

BA’s current SCADA system relies on Excel files that are fed through Open Platform Communications (OPC) servers. The OPC standard allows the production to access field information in real-time with greater flexibility and lower costs for the integration, development, and assembly of process automation or control systems [2].

However, as the Excel files have grown in size and are interconnected with other Excel files containing manually inserted data, the system has become slower and less effective. As a result of these challenges, a decision was made to introduce a new system, and PSS was acquired.

Currently, some *Circutor* equipment is already installed and feeding the SCADA with OPC. The ultimate objective is to enhance further and complete the installation of this system.

1.1 Context

1.1.1 BA Glass History

Barbosa & Almeida was incorporated in 1912 and dedicated to the commercialization of bottles. In 1930 the company began industrial activity in Campanhã with semi-automatic technology and changed its name to *Fábrica de Vidros Barbosa & Almeida, Lda* (Barbosa & Almeida Glass Factory LLC). After the introduction of automated technology and the use of automatic machines, production increased substantially, and in 1969 a new industrial unit in Avintes started operating with two regenerative furnaces (with heat recovery). In 1971 the first automatic Individual Section (IS) machine was installed, which allowed for a significant increase in the installed capacity. By

1979, the production relied on five IS machines, one of which was computerized. Currently, the Avintes plant has a production capacity of over 1 billion glass containers per year and employs around 300 people.

By 1993, BA acquired 94,5% of *CIVE - Companhia Industrial Vidreira, SA* from the state, a company located in Marinha Grande, with three furnaces. Later in 1995, CIVE merged by incorporation into BA. Through the years, BA acquired factories all around Europe: in Villafranca de los Barros (1998), in León (1999), in Venda Nova by incorporation of the *Sotancro* Group (2008), in Sierakow and Jedlice by acquisition of the Polish group *Warta Glass* (2012), in Gardelegen by acquisition of HNG Global (2016), and, lastly in Athens, Sofia, Plovdiv and Bucharest by the acquisition of *Yioula* Group (2017) [3].

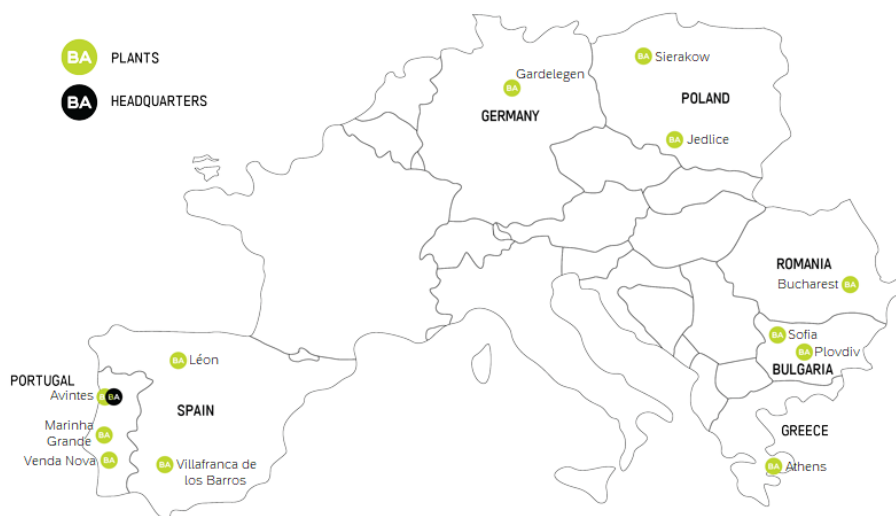


Figure 1.1: BA Glass's plants location

BA Group is composed of three divisions:

- Iberia (IB), formed by the plants of Avintes (AV), Marinha Grande (MG), Venda Nova (VN), León (LE) and Villafranca de Los Barros (VF), the first three located in Portugal and the last two in Spain;
- Central Europe (CE) consists of the plants of Sieraków (SI) and Jedlice (JE), located in Poland, and Gardelegen (GA), in Germany;
- Southeast Europe (SEE), composed of the plants of Athens (AT), located in Greece, Sofia (SO), and Plovdiv (PV), in Bulgaria, and Bucharest (BU), in Romania.

BA is currently present in over 70 countries with 3900 employees. The 12 plants combined manufacture over 11 billion glass containers annually for the food and beverage, pharmaceutical, and cosmetic industries. This results in almost 900 million euros turnover, giving BA a significant share of the world glass market [4].

1.1.2 Glass Manufacturing Process

The process of manufacturing a glass container starts with a value chain analysis to improve efficiency and reduce environmental impacts. The raw material for glass production is obtained from recycling used packaging. Recycled glass, also known in the industry as "cullet," makes up about 60% of a (green or amber) glass bottle and is a 100% recyclable material that can be used several times without losing quality or characteristics. Using cullet reduces the amount of raw materials required, the energy consumption for glass molting, carbon dioxide emissions, and the amount of glass deposited in landfills.

At the time of this study, only amber and dark green bottles were produced at BA Avintes, and the manufacturing process can be divided into six distinct phases with different technologies and particularities.

1. **Batch:** Other ingredients are added to the cullet: sand, with a high silicon content; limestone, as a stabilizer and sodium carbonate, to lower the melting point and save energy. The raw materials are stored in silos and later mixed.
2. **Fusion:** The resulting batch is conveyed through a network of conveyors (troughs) and introduced in the refractory furnaces. The temperature of the furnace can be controlled in real-time through the control panels of the molting furnaces. Once in the furnace, the glass batch is molten at 1500 to 1600 °C. From the homogenization of the raw materials to the output of molten glass, an average period of 24 hours occurs. The liquid glass flows to the production line through the refractory channels (or feeders).
3. **Forming:** The process begins with cutting the glass gob, which is launched by gravity and conveyed by channels and deflectors to the automatic molding machine. The gobs have the right amount of glass for the intended packaging model. Molding takes place in two phases. First, in the starting mold where the outline is formed. This is then transferred to the final mold, where it comes out with the final shape of the bottle. When they come out of the mold, the bottles are at approximately 600°C and are subjected to a very sudden temperature drop, creating internal stresses.
4. **Annealing and surface treatment:** To relieve these stresses, they are sent to the annealing lehr, and subjected to a new temperature rise to 650°C. Annealing takes approximately one hour. Then, cold treatment is applied to all bottles to make the glass more durable and prevent scratches.
5. **Inspection and quality control:** After annealing, the bottles are sent to the automatic inspection area (also called the cold zone), where inspection machines check all the bottles and detect defects at the visual, dimensional, thickness, and seams or cracks level.
6. **Cullet:** Bottles that do not meet the quality criteria are automatically rejected on the production line and are incorporated into the internal cullet circuit to be recycled again. Samples are also taken for laboratory analysis of other dimensional and mechanical resistance characteristics.

or

Packaging: The automatic palletization phase follows. Bottles are packed in layers and stacked on pallets for movement. They are then transferred to the packaging area, where the pallets are covered with a plastic film and labeled. From this moment on, the pallets are ready to be shipped to the end customer [3].

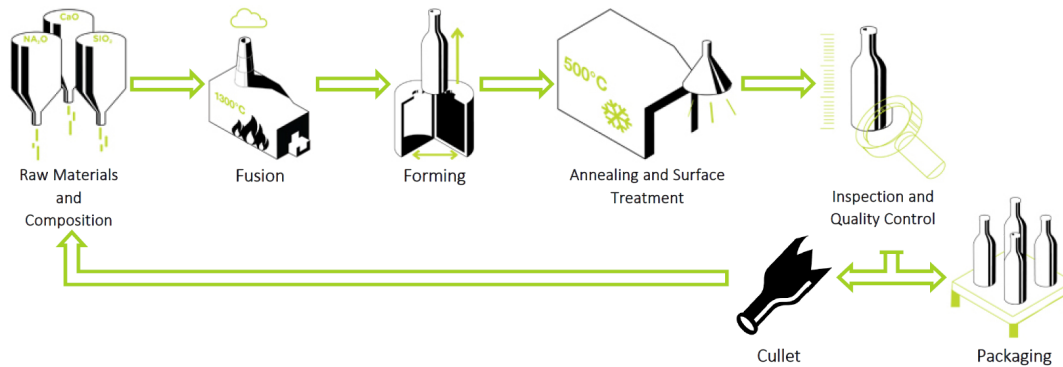


Figure 1.2: Glass container manufacturing process

The Avintes plant is comprised of three furnaces, namely AV2, AV4, and AV5, each with its own set of production lines. Furnace AV2 accommodates three production lines: L20, L21, and L22. Similarly, furnace AV4 houses three production lines: L41, L42, and L43. Lastly, furnace AV5 has four production lines: L51, L52, L53, and L54. These ten production lines collectively play a crucial role in the manufacturing operations of the Avintes plant, contributing to its overall productivity and output. Throughout this work, when referring to AV2, AV4, and AV5, it pertains to the furnaces.

1.1.3 Energy Sustainability in Industry 5.0

Industry 5.0 (I5.0), also known as the human-centric manufacturing era, builds upon the foundation laid by I4.0. While I4.0 focuses on integrating advanced technologies into industrial processes, I5.0 emphasizes the collaboration between humans and intelligent systems to achieve sustainable and inclusive manufacturing [5].

One of the key pillars of I5.0 is energetic sustainability, which aims to develop and implement energy-efficient practices to reduce the environmental impact of manufacturing processes. The technologies and methods developed in I4.0 are crucial in accomplishing this goal.

I4.0 has emerged as a transformative concept in the manufacturing and industrial sectors, enabling the integration of advanced technologies, such as the Internet of Things (IoT), Artificial Intelligence (AI), big data analytics, and automation, into industrial processes to create smart, connected systems.

In the context of concerns about the energy crisis, I4.0 has played a crucial role in addressing and mitigating energy-related challenges. Leveraging data and digital connectivity enables more

efficient and sustainable energy management practices across industries, while SCADA systems enhance monitoring and control capabilities.

I4.0 incorporates smart energy systems that utilize real-time data analysis to optimize consumption, reduce waste, and minimize energy usage during peak demand. IoT sensors and connected devices monitor and control energy-intensive processes, while AI-powered systems enable predictive and adaptive energy management for increased efficiency.

Furthermore, I4.0 facilitates the integration of renewable energy sources, also contributing to achieving energetic sustainability in I5.0. Smart grids and energy management platforms enable the seamless integration of renewable energy sources, reducing dependence on fossil fuels.

The collaboration between humans and intelligent systems, a fundamental aspect of I5.0, helps drive energetic sustainability. Humans can leverage the insights provided by I4.0 technologies to make informed decisions regarding energy usage, implement energy-saving practices, and continuously improve energy efficiency in manufacturing processes.

In summary, I5.0 builds upon the technological advancements of I4.0 and strongly emphasizes human collaboration and energetic sustainability. The technologies and practices developed in I4.0, such as data analytics, AI, and integration of renewable energy sources, will play a vital role in achieving the goals of I5.0, enabling sustainable and inclusive manufacturing processes that prioritize energy efficiency and minimize environmental impact.

1.2 Motivation

Energy is a critical input for glass manufacturing processes; consequently, the energy crisis has significantly impacted BA Glass.

The current global energy crisis has been primarily caused by a combination of factors, including the COVID-19 pandemic, extreme weather events (for example, the case of the summer of 2022's drought in Portugal), and supply chain disruptions caused by the war in Ukraine and conflicts in the Middle East [6]. These factors have led to a significant increase in the demand for energy and a decrease in the supply of energy resources such as oil, gas, and coal. As a result, energy prices have significantly increased.

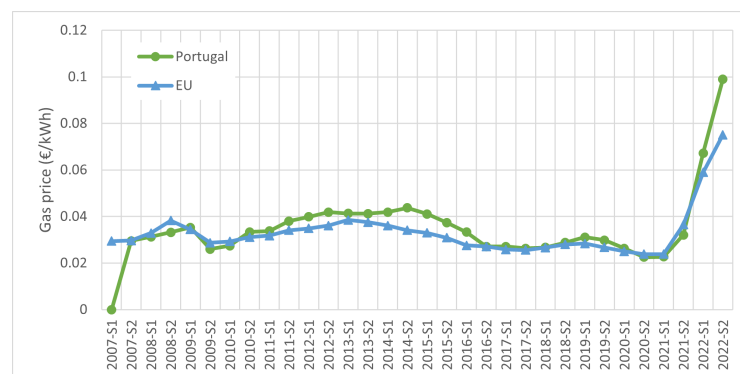


Figure 1.3: Evolution of gas prices for non-household consumers (source: EUROSTAT, 2023 [7])

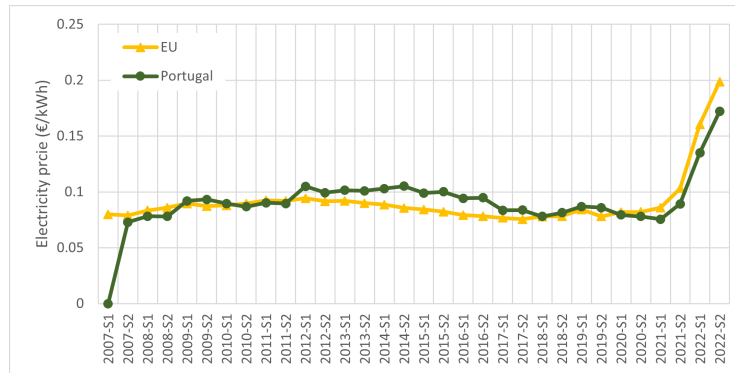


Figure 1.4: Evolution of electricity prices for non-household consumers (source: EUROSTAT, 2023 [7])

The Avintes facility is committed to sustainability, with a particular emphasis on reducing energy consumption and emissions through the increased use of recycled glass in its production processes. In addition, the plant has a cullet treatment station that serves its own needs and other plants within the BA group. Avintes has earned several certifications, including ISO 9001 for quality management [8], FSSC 22000 for food safety [9], ISO 14001 for environmental management [10], and SA 8000 for social accountability [11]. The facility is working towards achieving ISO 450001 for health and safety management [12] and ISO 500001 for energy management [13], further reinforcing its dedication to sustainable operations [3].

1.3 Problem Definition

In order to mitigate the impact of the energy crisis and reduce the carbon footprint that results from the plant's operations, it is essential for the Avintes plant to control its energy expenditure. Currently, data acquisition is performed manually or automatically (through OPC) in some areas, and all data is recorded in Excel. This process involves three files: one for manual readings, a second for automatic data linked to the first file, and a third one containing graphs and statistics used in daily meetings. However, the current use of Excel has resulted in slow and inefficient files, which highlights the need for a new system that can adequately meet the requirements of a company the size of BA Glass.

At the start of this study, a SCADA system is being implemented with *PowerStudio*'s software, which is a more advanced and efficient system. Currently, the system lacks the basic features of a SCADA, it's not intuitive for the user, and it lacks alarms. *PowerStudio* offers graph presentation and automatic report generation, but these functionalities are currently underutilized. Enhancing SCADA effectively addresses the challenges associated with analyzing large data files. *PowerStudio* enables users to visualize graphs and tables containing sensor values for any desired time period, providing a solution to the existing problem.

The Avintes plant requires significant enhancements in monitoring energy expenditure within its facility. With the data collected from PSS, this study endeavors to develop a precise consumption prediction tool using ML regression models.

1.4 Objectives

One of the objectives of this dissertation is to improve the SCADA system by enriching the Human Machine Interface (HMI), connecting missing sensors/meters, and configuring alarms to detect malfunctions. Additionally, a way of detecting flow meter malfunctions that lead to inaccuracies in the recorded data needs to be implemented. The purpose of these efforts is to contribute significantly to the energy monitoring of the plant.

To enhance the energy efficiency of BA even further, the second part of this dissertation will focus on the implementation and parameter determination of a machine learning (ML) tool that can accurately predict furnace energy consumption and evaluate targets. It aims to determine the optimal combination of training and prediction periods and select the most accurate supervised learning model using the available data. By analyzing these factors, the intention is to create a highly accurate consumption prediction tool for enhanced energy efficiency.

The research questions derived from this project are as follows:

1. Which ML regression model is the most accurate in predicting furnace consumption in the glass manufacturing industry? The first test scenario will address this question.
2. What is the optimal combination of training and testing data sizes for predicting consumption? The second test scenario will provide insights into this question.

As a notable outcome of this dissertation, an article based on this study's findings, "Energy Consumption Analysis in SCADA: A Case Study in the Glass Container Industry," has been prepared and is currently under submission for presentation at the Conference on Industry Science & Computer Science Innovation of 2023 [14].

1.5 Dissertation Structure

Besides the introduction, this dissertation contains five more chapters.

In chapter 2, state of the art is described, and related works are presented.

Chapter 3 focuses on the data collection process using *PowerStudio* and OPC Router and provides a comprehensive overview of implementing alarms. The SCADA devices in *PowerStudio* were used to monitor and record relevant variables, creating a comprehensive dataset.

Chapter 4 addresses the meticulous analysis following data collection that was conducted to identify abnormal values and ensure a normalized dataset for developing a regression model capable of predicting furnace consumption in Avintes. This chapter also concentrates on developing ML models for extrapolating predictive insights within furnace operations. These efforts contribute to improving operational efficiency and cost estimation in the Avintes facility.

Chapter 5 focuses on the testing and validation processes for the PSS data collection system and the prediction model developed for the furnaces' energy consumption. The data collection system is validated by comparing the values from flow meters on the plant floor with those recorded in SCADA. The prediction model undergoes extensive testing using historical data, assessing its performance across different time frames and training data sizes. Evaluation includes measuring accuracy and precision and assessing computational efficiency. Overall, the chapter provides insights into the reliability and effectiveness of the data collection system and prediction model, ensuring accurate data collection and reliable energy consumption forecasts for the furnaces.

Chapter 6 concludes the dissertation by presenting this project's results, limitations, and contributions and comments on future work.

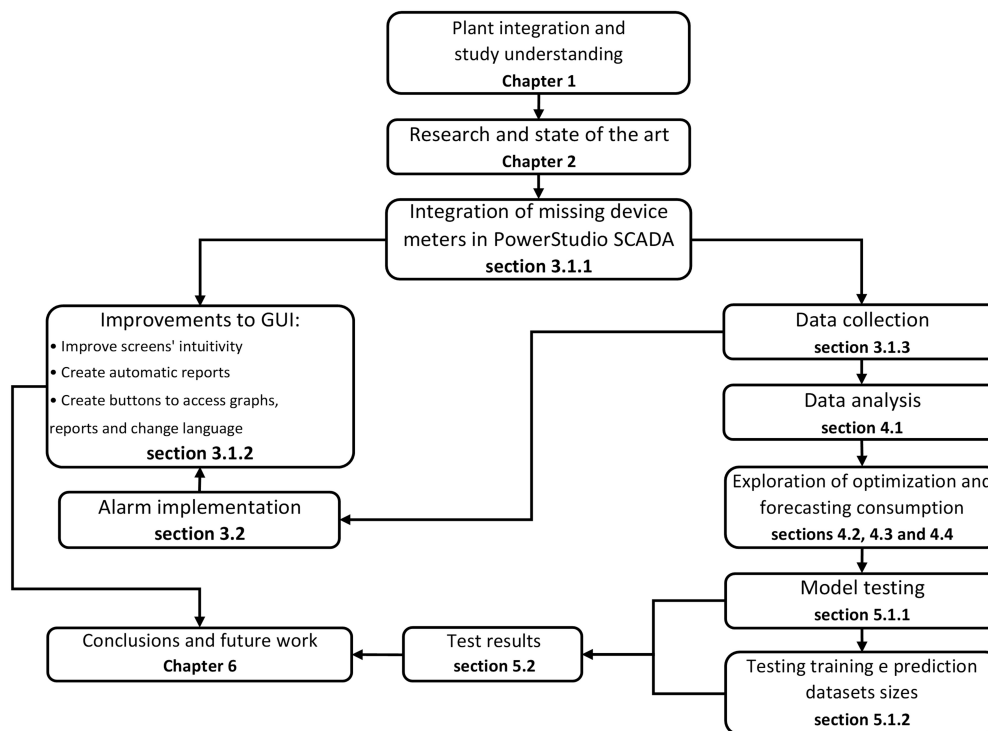


Figure 1.5: Workflow with document structure

Chapter 2

State of the Art

BA Glass' business model revolves around the efficient production of high-quality glass packaging, emphasizing innovation, exceptional customer service, sustainability, and global expansion. Given the continuous and ongoing nature of glass packaging manufacturing and the significant energy resources it entails, energy efficiency plays a crucial role in the company's priorities [3].

This chapter presents the state of the art related to this dissertation, providing a comprehensive theoretical framework. The literature review covers essential concepts related to SCADA and presents a range of effective methods for managing it, the role of OPC protocol as a bridge between legacy systems and *PowerStudio SCADA* (PSS), and the application of Machine Learning (ML) techniques and tools as a powerful means for optimizing energy efficiency in the plant.

This chapter lays this study's foundation and helps establish its context.

2.1 Supervisory Control And Data Acquisition System and Energy Monitoring

SCADA software applications collect data from sensors and actuators (S&A) and exchange control parameters with automation units, such as Programmable Logic Controllers (PLC). These applications display numerical and/or graphical information on the plant's behavior in real-time to an operator while also storing relevant variables for further analysis [1].

PLCs are the most commonly used data acquisition and process control devices in the industrial field. The server processes the data collected from the process, and the client (or viewer) is connected to the network with the server to access and communicate the data with the human operator. The servers are connected to the controllers through various communication drivers.

The monitoring and control function is a crucial aspect of SCADA systems. The primary functions of these systems include optimizing output parameters and efficiency by automatically controlling the technological process, displaying the real-time condition of the technological process, graphically displaying process data to develop efficient operating strategies, controlling process quantities logging, equipment condition, and alarm status effectively, generating periodic operating reports, allowing users to intervene directly in the process based on their access rights [2].

The utilization of networked smart energy meters for data exchange has been explored in previous works, specifically in [1] and [15]. In line with these studies, this dissertation aligns with the aforementioned research, examining the viability and benefits of an Ethernet-based network for integrating S&A and SCADA systems in the context of I4.0.

S&A are increasingly equipped with embedded Transmission Control Protocol/Internet Protocol (TCP/IP)-Ethernet ports. Consequently, adopting an Ethernet-based network becomes a viable option to connect the S&A network seamlessly with the SCADA system. This approach facilitates heterogeneity management and enables incorporating advanced S&A capabilities [1]. By leveraging the Ethernet-based network, the integration of S&A and SCADA systems can be achieved, ensuring smooth and efficient data exchange within the framework of I4.0.

In accordance with the work presented in the thesis [16], users' previous experience should be considered during the design process. Therefore the design of the new SCADA system should prioritize a graphical interface that closely resembles the existing system. This includes maintaining consistency in buttons, windows, and other visual components.

By replicating the graphical interface of the existing system as closely as possible, users can leverage their prior knowledge and experience, thereby minimizing the learning curve associated with the new system. This approach facilitates a smoother transition process, ensuring continued efficiency and effectiveness of industrial operations. In other words, a seamless transition between the existing and new systems allows users to quickly adapt and perform their tasks without significant disruptions or delays.

SCADA systems have been in use for several decades. In the early days, SCADA systems were mostly hardware-based and used analog S&A to control the process. Over time, SCADA systems have evolved to become more software-based by integrating digital technologies. The rise in the usage of microcontrollers and wireless technology led to the evolution of SCADA to Cyber-Physical Production Systems (CPPS). CPPS integrates physical and cyber systems to create smart, interconnected systems that can monitor and control industrial processes in real-time. CPPS combines various technologies, such as IoT, AI, cloud computing, data analytics, and ML, to create a new generation of industrial systems. CPPS can potentially increase the production's efficiency, enabling the flexible and re-configurable realization of automation system architectures [17, 18].

2.1.1 Digital Twin (DT)

The advances in the previously mentioned information technologies (cloud computing, big data analytics, and ML) enabled both Cyber-Physical Systems and Digital Twins. Consequently, CPS and DT emerged almost simultaneously. CPS and DTs belong to different categories, with CPS falling under the scientific category and DTs under the engineering category. In the context of industrial practices, DT technology offers significant benefits, allowing engineering systems to achieve higher levels of precision and better management [17, 18]. CPPS is a specialized type of CPS that integrates physical machinery and digital technology in manufacturing processes.

The DT typically comprises a physical entity, a virtual representation, and their data connections. It is increasingly being used to enhance the performance of physical entities by simulating

their behaviors and providing feedback through their virtual counterpart [17, 19]. Hence, DT technology is increasingly being explored as a potential solution for managing complex systems like SCADA or CPPS.

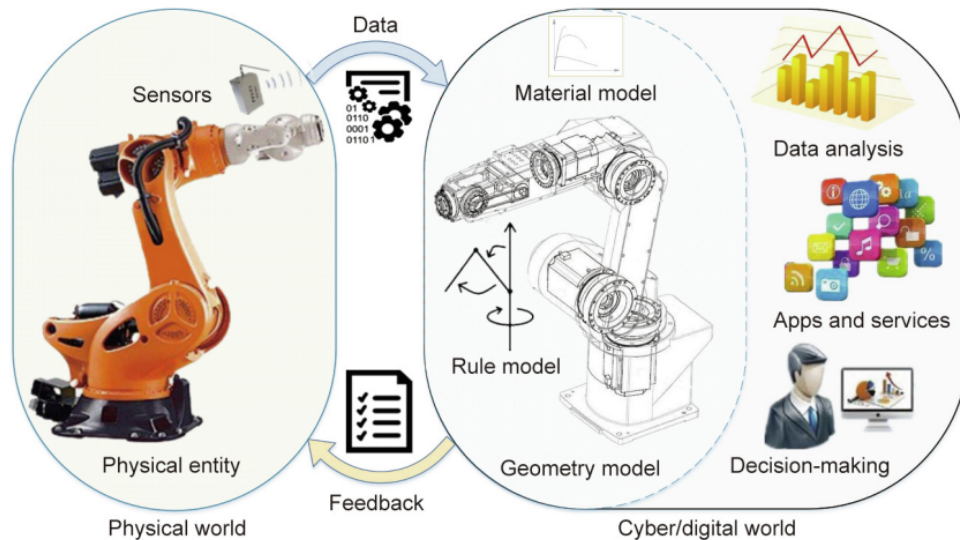


Figure 2.1: CPSs and DTs in manufacturing (source: Fei Tao *et al.*, 2019 [17])

2.1.2 DINASORE

The Dynamic INtelligent Architecture for Software MODular REconfiguration (DINASORE) is a novel framework that implements the industrial standard IEC 61499 using Function Blocks (FBs) in Python. It is designed to implement CPPS and SCADA management, aiming to provide standardized equipment integration with information systems and other platforms, including *PowerStudio* [20]. DINASORE utilizes the 4DIAC-IDE as a graphical user interface (GUI) to simplify the design and deployment of FBs for efficient and on-demand reconfiguration of the target equipment [21].

One of the key features of DINASORE is its seamless data integration with third-party platforms using OPC UA, enabling smooth communication and interoperability. This framework offers flexibility and reliability, making it suitable for various applications. However, it is important to note that as the number of FBs increases, the CPU and memory workload linearly escalates, which should be considered in large-scale implementations [21].

DINASORE addresses the challenges faced when using pre-existing algorithms in high-level programming languages, particularly in cases where equipment and software compatibility issues arise. By providing a standardized approach to CPPS and SCADA management, DINASORE and its companion technology, DT, offer a potential solution for integrating and managing heterogeneous systems, facilitating the seamless flow of information, and enhancing overall system performance.

2.1.3 PowerStudio SCADA

PSS is a software solution designed to centralize and manage information, process data and generate reports. Developed by *Circutor* to manage Electrical Energetic Efficiency (3E), *PowerStudio* is a robust factory and plant management tool. *Circutor* provides a wide range of devices that streamline the integration of any equipment in the factory, enabling the SCADA system to gather data more efficiently. *PowerStudio*'s SCADA delivers valuable data for the daily control of the plant floor enabling the management of different types of energy consumption, such as electricity, gas, and water. *PowerStudio*'s capabilities extend beyond energy management, providing a range of tools for energetic analysis, production ratios, network quality management, and energetic supervision over every factory equipment. *PowerStudio* uses OPC, a widely used standard in this field, to facilitate its connection to other software and hardware [22, 23].

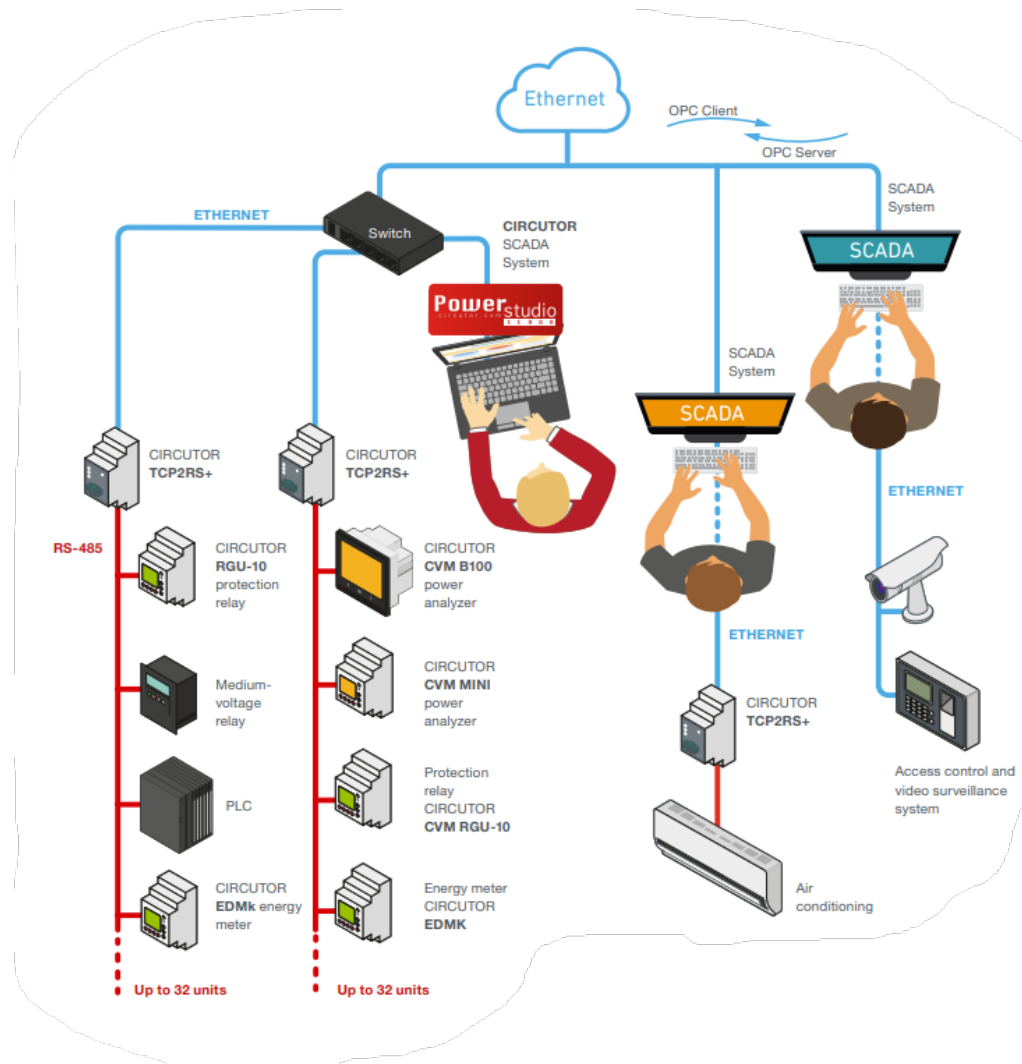


Figure 2.2: Illustration of *PowerStudio* SCADA OPC and Modbus communications (source: Circutor, 2016 [24])

2.1.4 OPC

OPC is an established standard recognized as the benchmark for secure and reliable data exchange in industrial automation and various other industries. The OPC standard comprises a set of specifications that govern the interactions between servers and clients from different vendors. Its inception dates back to 1996, driven by the objective of standardizing and unifying diverse PLC protocols such as Modbus and Profibus. By introducing a middle layer, OPC enables seamless communication between HMI/SCADA systems and these protocols, regardless of the specific manufacturer or vendor involved. This intermediary layer facilitates the translation of generic OPC read/write requests into device-specific commands and vice versa, ensuring interoperability across the industrial automation landscape [25].

Standard OPC servers offer several advantages for manufacturers' software clients that need to interface with various physical devices in their processes. By utilizing a standard OPC server, software clients can avoid the need to maintain a library of drivers specific to each device, simplifying the software architecture and reducing maintenance efforts.

End users also benefit from the use of standard OPC servers. One significant advantage is the cost reduction achieved by eliminating the need for proprietary device drivers. Instead, plug-and-play components from different suppliers can be seamlessly integrated, saving both time and resources during system setup and expansion. Additionally, relying on standard OPC components rather than specific drivers mitigates the risks associated with compatibility issues and potential software conflicts.

The OPC Data Access (DA) interface, provided by standard OPC servers, enables crucial functionalities such as real-time data extraction from various devices, including PLCs, DCSs, SCADAs, HMIs, and smart sensors. With this interface, software clients can efficiently read, write, and monitor process variables, facilitating effective control and monitoring of the manufacturing or industrial processes [26].

Manufacturers and end users can streamline their system integration processes by adhering to OPC standards and leveraging the flexibility of OPC servers. Real-time data extraction from various devices becomes seamless, reducing implementation time and ensuring compatibility across different components. Using standard OPC servers ultimately promotes interoperability, flexibility, and cost-effectiveness in industrial automation and control systems.

The OPC Foundation established OPC Classic in 1996 using Distributed Component Object Model (COM/DCOM) technology to facilitate information exchange between hardware devices. OPC Classic included OPC-DA for data exchange, OPC Alarm and Events (AE) for alarm and event information, and OPC Historical Data Access (HDA) for working with past data. However, due to outdated specifications, decreasing support for COM/DCOM, and the need for a unified service set, OPC Unified Architecture (UA) was developed as the next generation of OPC technology. OPC UA offers improved security, openness, and reliability, with advantages such as enhanced security measures, expanded transport options, and a comprehensive information model [2, 22, 26].

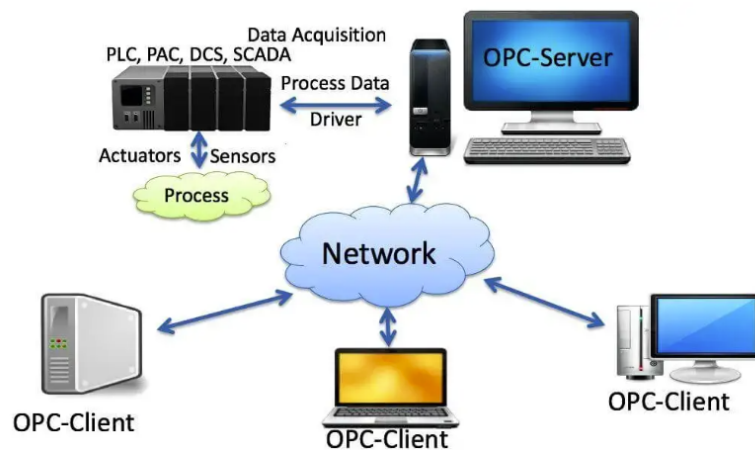


Figure 2.3: Illustration of OPC Client-Server communication (source: Instrumentation Tools, 2023 [27])

Several studies have highlighted the utilization of OPC for communication purposes in industrial control applications and SCADA systems. The following literature sources provide insights into the integration and communication capabilities facilitated by OPC:

- In Godoy and Pérez's study [1], OPC is employed as the communication protocol between the master PLC and the SCADA application.
- Nicola et al.'s study [2] showcases an example of OPC server-based application software that can be embedded within a SCADA system.
- Pereira, Reis, and Gonçalves's research [21] introduces DINASORE, which enables data integration with third-party platforms by utilizing OPC UA.
- Pinto's thesis [22] involves the development of a dashboard for energy consumption optimization. An external data module (OPC Router) is employed to input the necessary dashboard data automatically.
- Diaconescu and Spirleanu's work [26] employs OPC servers to communicate between industrial equipment and SCADA systems.
- Stefanov et al.'s work [28] focuses on SCADA modeling for performance and vulnerability assessment of integrated cyber-physical systems. Communication between the various components of the system is established through OPC.

In addition to the mentioned literature sources, it's important to highlight that *PowerStudio*'s software is an example that utilizes OPC for connecting to other software and hardware systems. Despite OPC-UA being a more modern and superior protocol, as evident from the referred literature, *PowerStudio* is not compatible.

2.2 Data-Driven Energy Analysis

The manufacturing and process industries face a significant challenge in effectively harnessing the potential of the increasing volume of recorded data. With the availability of affordable sensors

and the need for enhanced process monitoring and reporting, data is accumulating rapidly. This data explosion presents immense opportunities across various sectors, prompting enterprises to utilize ML tools to unlock their potential. The advent of I4.0 and the prominence of AI have further fueled the interest in ML tools, leading to a widespread drive in all industries to explore and leverage their capabilities [29].

ML can be categorized into three main groups: reinforced, supervised, and unsupervised. In unsupervised learning, ML algorithms are employed to analyze and cluster datasets that do not have predefined labels. On the other hand, supervised learning is an ML technique where an algorithm learns from labeled training data. It involves establishing a mapping between input variables and their corresponding output variables, using a labeled dataset to guide the learning process. The algorithm identifies patterns and relationships within the training data and utilizes this acquired knowledge to make predictions or classify new, unseen data. This study will focus on using supervised learning to predict the energy consumption of the furnaces in Avintes.

2.2.1 Regression-Based Supervised Methods

Regression-based supervised methods aim to model the relationship between inputs or independent variables and outputs. These models typically use parametric equations, where the parameters are estimated based on the available data. By explicitly capturing this relationship, these methods provide estimates of the association between individual inputs and the outcome. They also account for the influence of other inputs, allowing for adjusted measures of association [30].

2.2.1.1 Metrics

According to the work presented in the thesis [31], determining the appropriate metrics to utilize is an essential step in assessing a model's performance. This study selected the Coefficient of Determination (r^2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) as the initial performance metrics.

These metrics are widely used for evaluating regression models in predicting consumption because they provide different aspects of model performance. Considering these metrics together allows a better understanding of how well a supervised learning algorithm predicts consumption.

- r^2 (2.1): This metric measures the proportion of the variance in the dependent variable that the independent variables in the model can explain. It is calculated with the ratio between the Sum of Squares (SS) of residuals and the SS of the total. It ranges from 0 to 1, with higher values indicating a better fit of the model to the data. If r^2 is negative, then it means that the chosen model fits the data really poorly.

$$r^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^n (y_i - f(x_i))^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.1)$$

- MSE (2.2): This metric calculates the average of the squared differences between the predicted and actual values. It measures the average squared error of the model's predictions,

enabling a comprehensive assessment of overall prediction accuracy.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 \quad (2.2)$$

- **RMSE (2.3):** This metric is derived from the MSE by taking the square root of the average squared error. It measures the average magnitude of the errors in the model's predictions in the same units as the dependent variable. RMSE is more interpretable than MSE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2} \quad (2.3)$$

- **MAE (2.4):** This metric calculates the average of the absolute differences between the predicted and actual values. It measures the average absolute error of the model's predictions, providing a robust indicator of prediction accuracy.

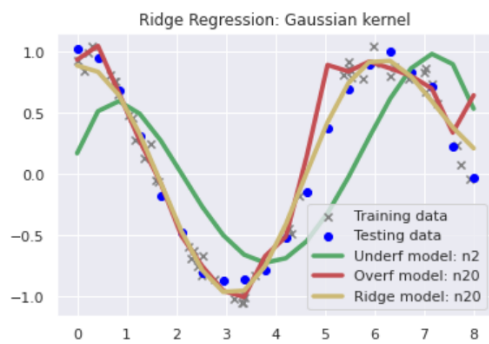
$$MAE = \frac{\sum_{i=1}^n |y_i - f(x_i)|}{n} \quad (2.4)$$

These metrics are commonly used to assess the accuracy and performance of regression models, with lower values of MSE, RMSE, and MAE indicating better model performance, while higher values of r^2 indicate a better fit of the model to the data [31].

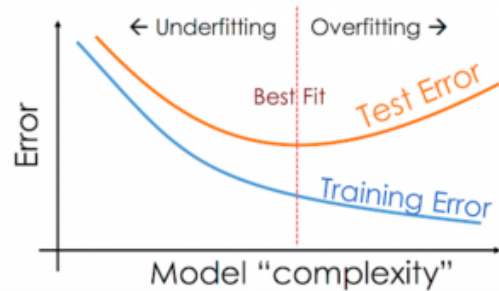
Testing and training errors are essential for evaluating a model's performance and generalization ability. The testing error measures how well the model performs on unseen data, while the training error assesses its fit to the training data. By comparing these errors, the following insights can be gained:

- High testing and training errors indicate underfitting, where the model fails to capture underlying patterns and generalize well.
- If the training error is significantly lower than the testing error, overfitting is present, meaning the model memorizes noise or specific details of the training data.
- When both errors are low and close in value, the model shows good generalization, successfully learning patterns and making accurate predictions on new data.

Specific error values depend on the problem, data complexity, and evaluation metrics used. Lower error values generally indicate better performance. However, it is important to compare errors across models and assess their relative performance for the specific problem domain.



(a) Underfitting and overfitting in simple regression ML (source: Aguiar *et al.*, 2022)



(b) Optimal zone for a good regression model evaluating testing and training errors (source: Kumar, 2023 [32])

Figure 2.4: Achieving a well-balanced model

As the referenced thesis [31] explains, selecting appropriate performance metrics, including the r^2 , MSE, RMSE, and MAE, conducting a split between train and test sets, and visual analysis of the graphical outputs enables a comprehensive evaluation of the model's performance. The combination of quantitative metrics and graphical outputs provides a holistic assessment of predictive accuracy, allowing for informed decision-making and potential improvements in the model.

2.2.1.2 Decision Tree Methods

Decision trees are constructed by recursively partitioning the input space into hypercubes to create regions with relatively homogeneous outcomes. This process involves applying binary splitting rules on the input variables hierarchically. The algorithm explores all possible binary splits and selects the one that maximizes the distinction between the output values of the resulting groups. While trees can be grown until they achieve purity in the terminal nodes, it is generally discouraged as it can lead to unstable estimates.

Once a decision tree is built using training data, predictions for new observations are made by traversing it based on their input values and determining the most frequent class (for classification) or the mean of outcomes (for regression) in the corresponding terminal node. Decision trees require optimization regarding the number of variables and the depth of the tree. Typically, prediction error, such as misclassification rate for classification or mean-square error for regression, is used as the criterion for optimization, often through cross-validation.

Decision trees offer advantages such as interpretability, accommodating various types of predictors, and scalability to large datasets. However, they exhibit high variability, making them sensitive to slight changes in the data, which can result in different splitting rules and tree structures. Additionally, decision trees tend to overfit the training data, leading to a suboptimal performance on external test sets. Techniques like bagging can be employed to mitigate these issues, which involves averaging multiple trees trained on resampled versions of the training data [30].

2.2.1.3 Regularized Regression Methods

In the context of supervised learning, where the goal is to predict an output variable based on input features, feature selection becomes a crucial step in building effective models. Regularized regression methods, such as ridge regression and the Least Absolute Shrinkage and Selection Operator (LASSO), offer valuable approaches for feature selection.

Ridge regression and the LASSO both introduce constraints on the coefficients of the features in the model. Ridge regression restricts the SS of the coefficients (L2 norm) (Equation 2.5), while the LASSO restricts the Sum of Absolute Values (SAV) of the coefficients (L1 norm) (Equation 2.6). These regularization techniques encourage the model to assign smaller weights or even zero weights to less informative or redundant features, promoting sparsity and improving the interpretability of the model.

$$SS_{ridge}(\beta) = \sum_i (y_i - \beta_0 - \sum_{j=1}^k x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^k \beta_j^2 \quad (2.5)$$

$$SAV_{LASSO}(\beta) = \sum_i (y_i - \beta_0 - \sum_{j=1}^k x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^k |\beta_j| \quad (2.6)$$

By controlling the regularization parameter λ , practitioners can control the degree of shrinkage applied to the coefficients. A larger λ value results in more coefficients being forced towards zero, effectively selecting a subset of features that contribute the most to the model's predictive power. This provides a way to automatically determine the relevant features and mitigate the risk of overfitting.

Extensions of the LASSO, such as the group LASSO, have been developed to handle scenarios where certain groups of features are expected to play a role together. This allows for a joint selection of related features.

While regularized regression methods like the LASSO and its extensions may not always guarantee consistent variable selection, they have demonstrated their usefulness and have been widely applied in various domains, including genetics and genomics. These methods enable researchers to uncover the most relevant features and improve model performance [30].

2.2.2 Data Science Programming Languages

Data science programming languages provide the tools and libraries needed to extract insights from complex datasets. With its user-friendly syntax and extensive libraries, Python has become immensely popular for data manipulation, analysis, and ML. R, known for its statistical analysis capabilities, is favored by statisticians and researchers. Julia offers high-performance and parallel computing, making it suitable for large-scale data processing. SQL is crucial in working with databases and efficiently querying structured data. Python stands out among these languages due to its versatility, rich ecosystem, and widespread adoption in the data science community.

Python is a popular high-level interpreted programming language known for its user-friendly nature and suitability for various projects. It was created by Guido Van Rossum in 1991 and has since evolved to its current version, Python 3. Python has gained significant traction in ML and Big Data due to its extensive collection of third-party libraries catering to these domains. These libraries provide potent tools and algorithms for data analysis, modeling, and visualization.

One of the notable features of Python is its package management system, with "pip" being the standard package manager used to install and manage libraries. This allows users to incorporate and leverage various specialized libraries within their projects efficiently.

In the context of scientific applications, Python has proven to be a valuable tool. Its versatility and extensive library ecosystem make it well-suited for various scientific disciplines, including physics, biology, chemistry, and engineering. Scientists can leverage Python's capabilities to process and analyze large datasets, create visualizations, implement advanced statistical models, and facilitate collaborative research efforts.

Some notable libraries commonly used in scientific applications include:

- *NumPy*: A fundamental library for numerical computing in Python, providing efficient array operations and mathematical functions [33].
- *Pandas*: A powerful data manipulation and analysis library offering versatile data structures and tools [34].
- *Matplotlib*: A comprehensive plotting library that enables the creation of static, animated, and interactive visualizations [35].
- *SciPy*: A library that provides a wide range of scientific and mathematical algorithms, including optimization, linear algebra, signal processing, and more [36].
- *Scikit-learn*: A popular ML library that offers a comprehensive set of tools for various tasks, such as classification, regression, clustering, and dimensionality reduction [37].
- *TensorFlow* and *PyTorch*: Deep learning frameworks widely used for building and training neural networks [38, 39].

These libraries and many others contribute to Python's reputation as a versatile and powerful language for scientific research and data-driven applications [40].

2.2.2.1 *Pandas*

The Python Data Analysis Library, commonly known as "*pandas*", was developed in 2008 and has gained immense popularity as one of the most widely used Python libraries. Its success can be attributed to the introduction of powerful data structures such as Series and DataFrames.

Pandas' DataFrame, in particular, has become a go-to choice for data manipulation and analysis tasks. With its spreadsheet-like structure comprising rows (entries) and columns (attributes), DataFrames provide a flexible and intuitive way to handle and analyze tabular data. This tabular representation allows for easy indexing, slicing, filtering, and transformation of data, making it a valuable tool for exploratory data analysis and preprocessing.

One of the factors contributing to pandas' popularity is its integration with *NumPy*, another widely used library for numerical computations. By leveraging the efficient numerical operations provided by *NumPy*, pandas can perform quick and optimized calculations on large datasets. This combination of pandas and *NumPy* forms a powerful toolkit for data processing and analysis in Python, ensuring optimal performance and accuracy in developing prediction models for this study.

Throughout this work, when referring to DataFrames, it pertains to the pandas object. The rows within a DataFrame represent individual entries, while the columns correspond to different attributes or features associated with the data.

2.2.2.2 *Scikit-learn*

The *scikit-learn* library is an indispensable asset in any ML Python toolkit. It provides a comprehensive collection of popular models for classification, regression, and clustering tasks and a wide range of tools for preprocessing and model evaluation.

Scikit-learn offers various algorithms, including decision trees, support vector machines, random forests, gradient boosting, and neural networks. These algorithms cover a broad spectrum of ML tasks, empowering practitioners to tackle various problems across different domains [37].

To assess and validate the performance of ML models, *scikit-learn* provides robust evaluation metrics and techniques. Cross-validation, grid search, and model selection tools assist in optimizing hyperparameters and selecting the best-performing models. These evaluation tools aid in building reliable and generalizable models.

The popularity of *scikit-learn* can be attributed to its user-friendly and well-documented application programming interface (API), which facilitates easy integration into ML workflows.

Overall, the *scikit-learn* library plays a vital role in enabling researchers, data scientists, and practitioners to effectively apply ML techniques to their datasets, promoting innovation and advancing the field of ML.

2.2.3 Hyperparameter Optimization

ML models often rely on a diverse range of input variables, each with its own specific range of values. *GridSearchCV* is a powerful technique used for hyperparameter tuning, which aims to find the optimal values for the hyperparameters of a given model. The choice of hyperparameters can significantly influence the performance of a model. However, determining the best values for hyperparameters beforehand is challenging. *GridSearchCV* automates the process by exhaustively searching through a specified dictionary of hyperparameters and their possible values.

In *scikit-learn*'s "model_selection" package, *GridSearchCV* provides a convenient way to iterate over the defined hyperparameters and fit the model on the training set. It systematically evaluates the model's performance for each combination of hyperparameter values. Finally, it allows us to select the best set of parameters based on the evaluation results.

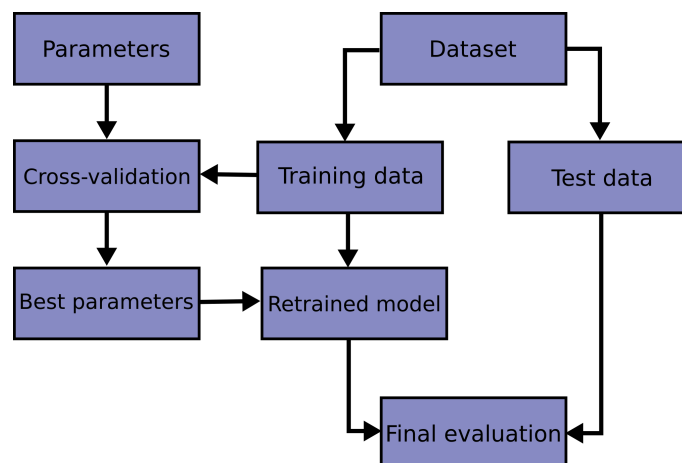


Figure 2.5: Grid search workflow (source: Pedregosa *et al.*, 2011 [37])

By utilizing *GridSearchCV*, we can save a significant amount of time and resources that would otherwise be required for manual tuning. It streamlines the hyperparameter optimization process and helps achieve the best possible performance for the model [41].

2.2.4 Transfer Learning

Supervised ML techniques have proven effective in various applications. However, their performance relies on the assumption that training and test data share the same features and distribution. Obtaining high-quality labeled training data can be challenging and costly, limiting the practical applicability of these methods.

To address this issue, active learning reduces annotation effort by designing an active learner to query unlabeled instances for labeling. An active learner can achieve high accuracy with fewer labeled examples by selecting informative data points. However, active learning methods often assume a budget for querying labeled data, which may be limited in real-world scenarios.

In contrast, transfer learning enables training and testing on different domains, tasks, and distributions. By leveraging knowledge or labeled data from related fields, an ML algorithm can improve performance in the target domain. Transfer learning offers an alternative approach to learning models with minimal human supervision compared to semi-supervised and active learning methods. It is beneficial when training data is scarce or costly to collect for each specific domain.

Transfer learning finds applications in various domains, such as recognizing apples to aid in identifying pears or learning to play the electronic organ to facilitate learning the piano. It allows for the reuse of training data or extracted knowledge from related domains, enabling the development of precise models for the target domain. In scenarios where collecting sufficient training data is impractical, transfer learning becomes a desirable and crucial approach for knowledge transfer between tasks or environments [42].

Chapter 3

PowerStudio Data Collection and Alarmistics

This chapter focuses on the data collection process using *PowerStudio* and OPC Router. The subsequent data analysis is explored in Chapter 4.

Through PSS's devices, various relevant variables were monitored and recorded, providing a comprehensive dataset for analysis. This dataset was the foundation for developing the ML model capable of predicting the consumption of Avintes's furnaces.

Lastly, this chapter covers the implementation of alarms in PSS, enabling real-time monitoring and alerting for critical events and abnormal conditions within the plant operations. This ensures prompt detection and response to potential issues, enhancing operational reliability.

3.1 Integration of *PowerStudio* SCADA and OPC Router's Modular System

This section delves into the integration and communication of devices within the SCADA system and the development of the external data analytics module employed in this study.

The integration and communication of devices within the SCADA system were paramount for this research. Through seamless integration, various devices, such as sensors, input/output devices, and converters, were effectively connected and coordinated, enabling the collection and transmission of real-time data. This integration facilitated a comprehensive view of the operational processes and allowed for efficient monitoring and control of critical parameters.

Furthermore, an external data analytics module was specifically developed to augment the capabilities of the SCADA system. This module collects data and allows connections to plug-ins facilitating informed decision-making and process optimization.

This module provides a powerful means to analyze and interpret the vast amounts of data the SCADA system generates. Its integration with SCADA created a holistic and intelligent system, enhancing the overall performance and effectiveness of the study.

In summary, the integration and communication of devices within the SCADA system and the data analytics module's development played a pivotal role in this study. Together, they facilitated comprehensive data analysis, empowered decision-making, and contributed to the overall success and effectiveness of the research.

3.1.1 Smart energy meters and IoT

The energy meter is classified into three types. They are electromechanical meters, electronic meters, and smart energy meters. This innovative smart device is essential for efficiently reviewing and controlling industrial equipment across various industries and reducing production costs. By accurately measuring electrical parameters and utilizing universal timestamps in the transmission system, we can precisely predict measurement precision, isolate faults, and detect issues.

One notable advancement in this field is the digital electronic meter, which boasts a high resolution, improved efficiency, and compatibility with low current and voltage operations. Its user-friendly features, such as easy reading and installation, make it a preferred choice. Moreover, the integration of a Global System for Mobile Communications (GSM) modem¹ allows for continuous monitoring of the electrical power supply without the need for human intervention.

The electronic energy meter has been designed using IoT principles and a GSM module to enhance its capabilities further. As a comprehensive network of sensing and communication devices, IoT enables the control of various quantities. This meter is particularly suitable for industrial and household applications, as it can accurately measure the energy consumption of individual electrical equipment without disrupting their current operation.

Conventional analog and electronic meters, which have been in use since the early stages, rely on manual operation. A meter reader is responsible for recording the readings, which are then used to generate billing information.

Overall, the selected smart energy meter based on IoT technology exhibits remarkable efficiency and can seamlessly measure energy consumption in households and industries, providing valuable insights for optimizing energy usage [43].

In BA Avintes, the readings of sensors and flow meters are carried out using *Circutor's* devices. These devices include the LM50+, impulse counters equipped with 50 slots, the LM4A for analog sensor readings, and the TCP1RS+, which converts serial communications standards RS-232 and RS-485 to Ethernet through TCP. The connection is established by connecting Ethernet cables from each TCP converter on different floors to the network hub located on the respective floor. To ensure proper network configuration, free IP addresses from BA's intranet are assigned to each device which are then configured with the MAC addresses as indicated on the devices [44]. For equipment that doesn't allow the connection of *Circutor's* devices, *PowerStudio* allows the configuration of generic Modbus TCP devices. With this type of generic device, the user only needs to configure the IP address of the equipment that needs to connect to *PowerStudio*. The communication of the devices with SCADA is shown in Figure 3.1.

¹A GSM modem is a specialized modem that functions with a Subscriber Identification Module (SIM) card and operates through a subscription with a mobile operator, similar to a mobile phone.

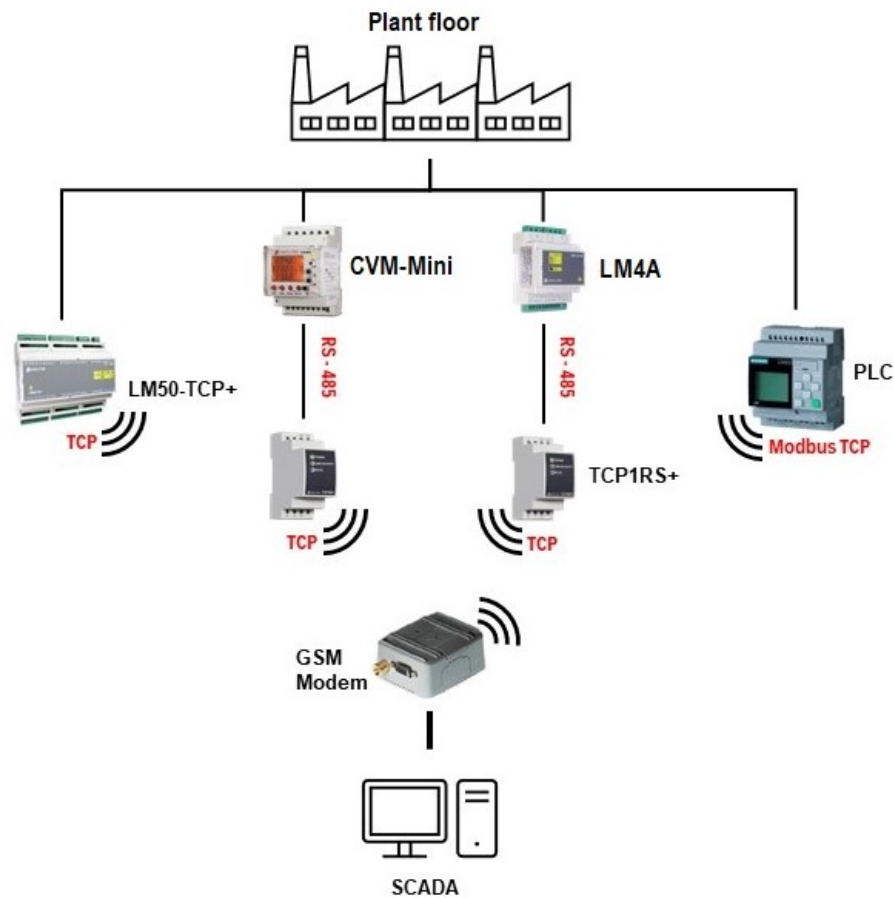


Figure 3.1: SCADA devices communication scheme

3.1.2 Graphic User Interface

Improvements to the Graphic User Interface (GUI) encompass several key enhancements to enhance user experience and functionality. To cater to BA's international presence, the screens should be designed to display content in both Portuguese and English, accommodating users from different regions. Additionally, implementing an automated report generation feature would streamline the reporting process, saving time and effort.

The interface is now more intuitive and user-friendly, focusing on presenting significant values prominently on the screens, the values of each flow meter, and the instantaneous consumption. This ensures that users can quickly and easily access important information. Furthermore, incorporating buttons to open graphs displaying the values of each variable adds a visual element to the interface, enabling users to analyze data trends and patterns more effectively. Moreover, a dedicated screen shown in Figure 3.2 has been developed for workers to conveniently modify the maximum and minimum limits of alarms. This screen greatly simplifies the process of adjusting variable setpoints, allowing for seamless adaptation to changing conditions.

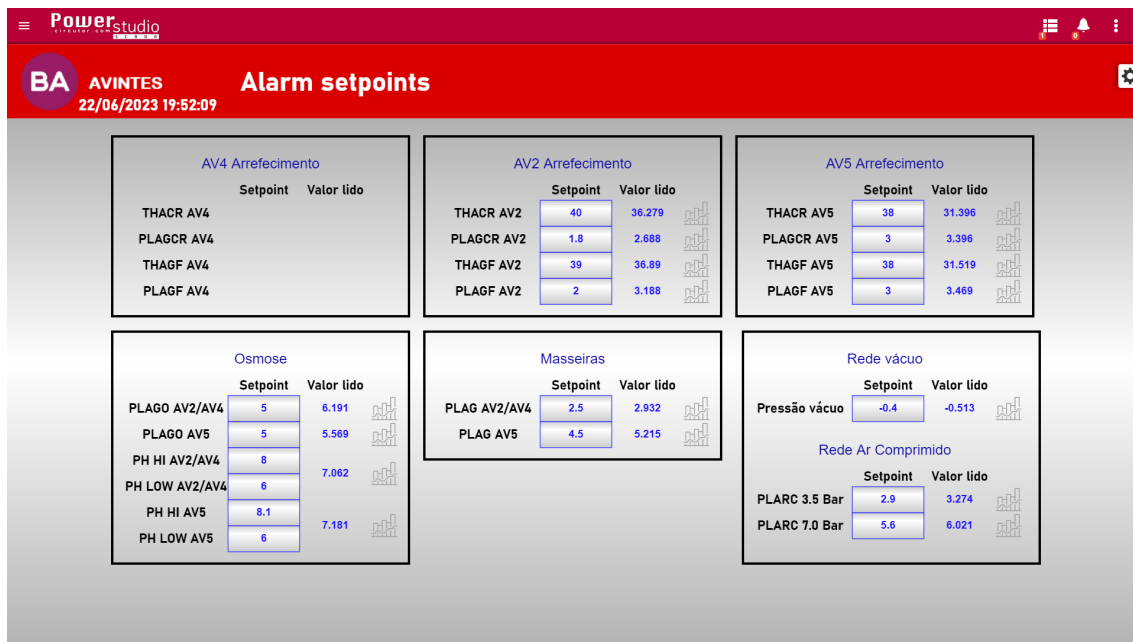


Figure 3.2: Alarm setpoint definition screen

The main screen provides a comprehensive overview of consumption metrics, presenting monthly, daily, and instantaneous consumption values, accompanied by an analog bar indicating whether the plant is operating within the predefined target range or not. This visual representation provides users with immediate feedback on the plant's performance in relation to the set targets.

By implementing these GUI improvements, BA can enhance user accessibility, streamline reporting processes, improve data analysis capabilities, and provide a more comprehensive and user-friendly interface for effectively monitoring and managing furnace operations.

Appendix A.1 presents print screens with the before and after improvements.

3.1.3 External Data Analytics Module (OPC Router)

Before the start of this dissertation, a "connection" between PSS and an Excel file was already implemented to automatically input data from the plant that, up until that point, was acquired manually. Each variable within a calculated variable group in PSS (OPC-TAG) created in the Editor program is assigned to the device input associated with the respective flow meter. This is achieved with an intermediate program, OPC Router. Whenever a new device is added to SCADA, it is necessary to update "OPC Devices" with *PowerStudio's* OPC Server Setup program and restart OPC Server to ensure the changes take effect. An OPC Classic server configured with the computer running *PowerStudio* IP address facilitates communication between the PSS and the OPC Router.

Within the OPC Router interface, a connection between the OPC-TAG group and the Excel file is established through the transfer objects "OPC Data Access" and "Excel," respectively. OPC Data Access is configured with the previously mentioned OPC Classic server and the "tag browser" feature to choose the appropriate variables from the OPC-TAG group. The Excel transfer object

was configured, specifying the target file as "*Leituras Diárias 23:59*" (where the values will be transmitted) and the specific cells to which each value should be sent.

Lastly, a time trigger was configured to automatically input the data daily at 23:59, ensuring regular and timely updates to the Excel file. The file's contents are used for data analytics and the energy consumption dashboard.

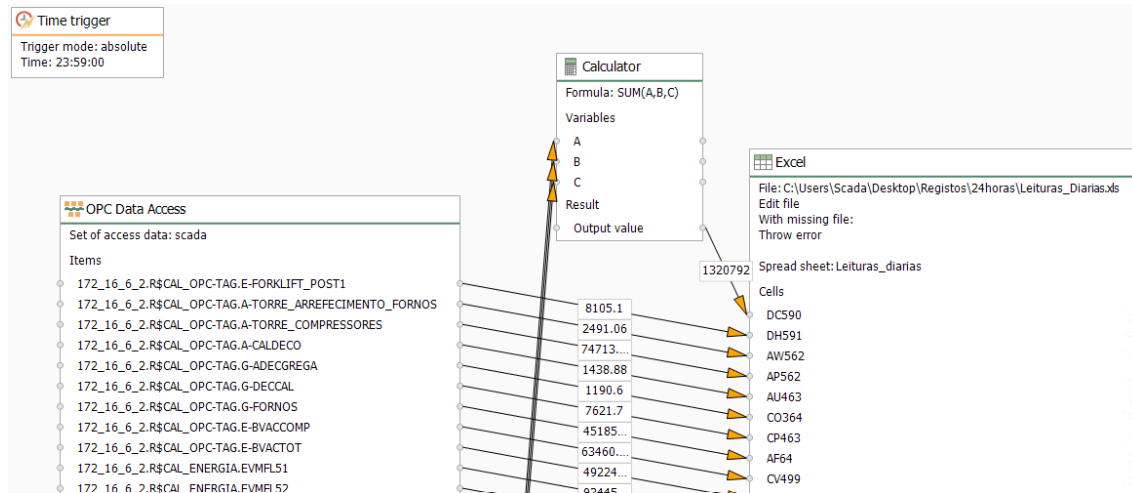


Figure 3.3: Connection created in OPC Router to register SCADA's values in Excel

The connections were expanded and enhanced throughout this study to encompass a broader range of devices equipped with sensors and flow meters. These augmentations were implemented with the overarching goal of optimizing the functionality of the SCADA system.

3.2 Integration of Alarmistics and SCADA

The initial hurdle that required surmounting involved updating *PowerStudio* to the most recent iteration, version 4.29.1. The existing version utilized in Avintes, namely 4.0.13, presented certain limitations. Notably, it could not acknowledge alarms within the browser client and took a long time during the application export process. Facilitating the update necessitated the creation of an exception within the firewall configurations, specifically for the computer's own IP address.

A significant challenge in the current SCADA system is the absence of a reliable means to detect potential malfunctions in the flow meters. Consequently, this leads to occasional inaccuracies in the recorded data, and identifying such discrepancies becomes exceedingly arduous.

The course of action for integrating flow meter fault detection alarms is outlined as follows:

Within the PSS Editor application, numeric calculated variables (that will hold the values 0 or 1) were added in a group created and labeled "*Alarmes Contadores*" (Flow meter Alarms in English). These variables shall subsequently be employed in configuring events that activate the corresponding alarms. The elaboration of the mentioned events depends on whether the variables are activated (with a value of 1) or deactivated (with a value of 0).

In *Circutor's PowerStudio* OPC Server Setup application, the inclusion of the newly formed group of calculated variables from "*PowerStudio* devices" into the "OPC Devices" category is necessary. Restarting the OPC Server is advised subsequent to this action.

A script in C# was created after accessing the plugin menus when launching the OPC Router interface. This script was specifically designed to ascertain the disparity between the previous and current values of the data tag (input). When this discrepancy exceeds a predefined threshold, set the output variable (alarm) to true; otherwise, assign it a false value. This script shall also equate the current value to the preceding value. Refer to Appendix C.1 to consult the script. It is crucial to note that each threshold was determined based on analyzing the corresponding input from each flow meter.

Then, a connection was established with the OPC Data Access transfer object (Figure 3.4) for transmitting the input to the script. The script's output was then connected to another OPC Data Access transfer object encompassing the variables sourced from the "*Alarmes Contadores*" group.

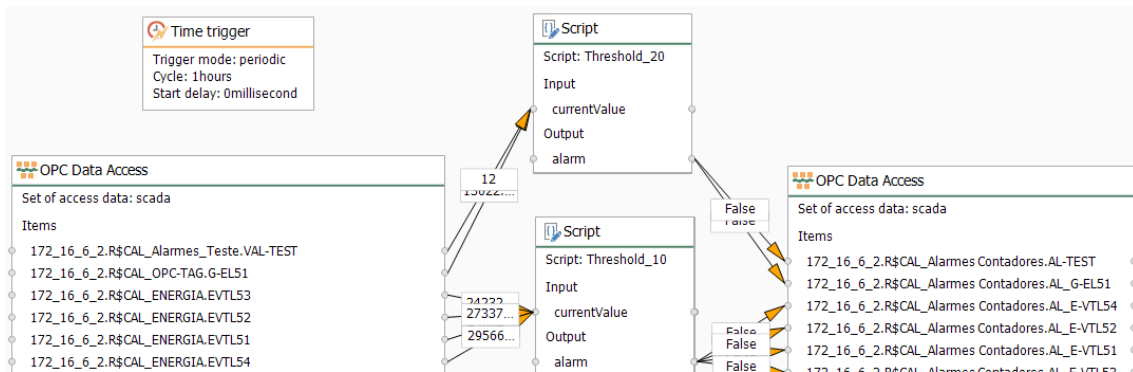


Figure 3.4: Connection in OPC Router to trigger alarms when a flow meter malfunction is detected

Chapter 4

Data-driven Energy Management, Monitoring and Forecasting

Following data collection, a thorough analysis was conducted to identify abnormal values, ensuring a normalized and consistent dataset for the models to learn from.

Integrating data collection, analysis, and ML techniques demonstrates a comprehensive approach to harnessing data-driven insights for enhancing overall operational efficiency.

Furthermore, this chapter discusses the development of ML models specifically designed to predict the energy consumption of Avintes's furnaces. These models leverage historical data and the planned pull to forecast future energy consumption accurately for the following week.

In addition to predictive modeling, this chapter explores optimizing energy consumption within the furnaces. This involved analyzing various operational parameters to maximize efficiency. The furnaces can operate more effectively by implementing optimized settings while reducing energy costs and environmental impact.

Lastly, the chapter delves into estimating energy consumption for similar furnaces using the predictive model developed for one specific furnace. This approach allows for the extrapolation of insights from one furnace to similar setups, providing valuable guidance for energy management and resource allocation.

4.1 Dataset Analysis

As explained earlier in this dissertation, a lot of data has already been registered, and the data log contains data from 2016 but more reliably from 2021. The Excel file "*Registos consumos_auto_1*" (Figure B.1) is where most data were extracted for this study. This file contains several sheets with readings (both manual and automatic) of the gas and electric energy consumption of the furnaces, refiners, feeders, lehrs, mold lehrs, and shrinking machines. It also contains recorded data from each furnace and production line's pull¹ (calculated with Equation 4.1). In this file, the specific

¹Amount of glass produced in a day

consumption² of each furnace and equipment of the production lines are also calculated, as shown in Equations 4.2 and 4.3. This value represents a more straightforward way of measuring the efficiency of the plant floor.

$$Pull[kg] = Weight_{bottle}[g] \times Velocity[bottles/min] \times \frac{480}{1000} \quad (4.1)$$

$$PCI_{gas}[kcal/m^3] = PCI_{gas}[kWh/m^3] \times 860.0506 \quad (4.2)$$

$$Consumption[kcal/kg] = V[m^3] \times \left(\frac{p_{atm}[kPa] + p_{func}[kPa]}{p_{atm}[kPa]} + \frac{T_{atm}[K]}{T_{func}[K]} \right) \times \frac{PCI_{gas}[kcal/m^3]}{Pull[kg]} \quad (4.3)$$

The molting process of the glass mixture involves two main components: gas combustion and electric boosting. This study aims to predict the total energy consumption of each furnace and determine the optimal combination of electric boosting and gas usage while considering the furnace infrastructure limits.

The dataset is comprised of labeled features. Firstly, the recorded date provides a temporal context for the data. Additionally, the dataset contains measurements of pull, represented in kilograms, for each production line and furnace. Cullet, expressed as a percentage, represents the proportion of recycled glass material (cullet) used relative to the total quantity of cullet and raw material utilized. The dataset also includes boosting, measured in kilowatt-hours (kWh), which denotes the electric energy consumed by each furnace's boosting and production line machinery. Gas consumption for each furnace and production line machinery is captured in units of normal cubic meters (Nm³), providing insights into the amount of gas utilized. Finally, the dataset records the PCI (kcal/m³), which stands for the lower calorific power. In order to guarantee the dependability and uniformity of the data, information prior to 2022 was excluded from the dataset. This choice was based on the availability of more reliable data from 2021, and any discrepancies arising from maintenance activities on the furnaces in 2021 had been resolved by 2022.

Several variables influence the fusion process. A higher cullet percentage in the batch mixture leads to decreased consumption, as glass with a higher cullet content is easier to melt than forming new glass. The color of the glass also plays a factor in the molting process. In Avintes, variations of amber and green glass are the ones produced. The quantity of cullet inserted in the batch mixture affects the color of the glass. Certain colors may facilitate molting due to a higher cullet percentage. Additionally, energy prices play a crucial role in determining the optimal mix of electric boosting and gas usage. When electric energy is cheaper, increasing electric boosting becomes more cost-effective.

Furthermore, different furnaces with varying ages require specific considerations for electric boosting. Boosting can elevate the temperature of the furnace crown, and older furnaces such as AV2 have a weaker infrastructure. Therefore, it is essential to exercise caution with boosting in older furnaces.

²Consumption value in kcal/kg of molten glass

A CSV file was utilized to access previously mentioned variables but was unavailable in the "Registros consumos_auto_1" file. This file was extracted from BA's software BAMEX³ and contains information regarding the quantity of raw material used in the batch mixture, the glass color, and the log of furnace crown temperature. It is important to note that this file only includes data until October 2022.

The furnace data exhibits significant fluctuations, including depressions and spikes. These variations can be attributed to production line halts, inaccurate value readings, or sudden changes in the pull. It is important to note that the pull and consumption are closely linked; as more glass is molten, the furnaces consume more. In BA, the pull for the upcoming week is planned and recorded in an Excel file named "Tiragens" (Pull in English) weekly (Figure B.2).

A plot was created to visualize both data sets to assess the similarity between the actual pull during production and the planned pull. The purpose was to determine if the planned pull could effectively filter the actual pull data for training the models. By comparing the two plots, we aimed to infer if there is a strong similarity between the actual and planned pull. If a close resemblance is observed, it would suggest that the planned pull can serve as a reliable filter for the actual pull data.

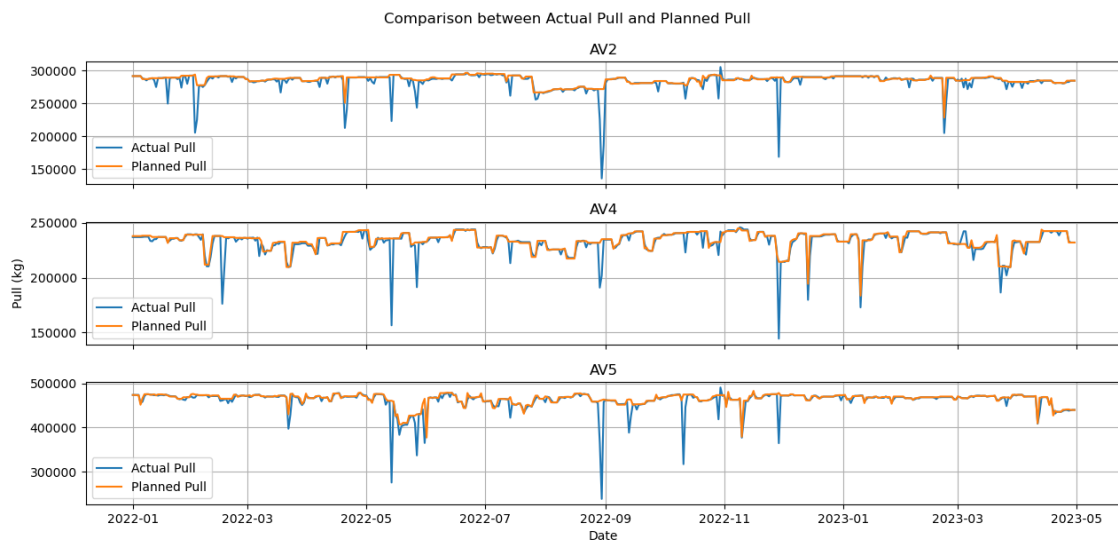


Figure 4.1: Comparison between actual pull and planned pull for each furnace

Upon examining Figure 4.1, a promising approach for filtering the pull data is to utilize the planned pull as a benchmark for identifying and excluding abnormal data points. To achieve this, an in-depth analysis of this file was conducted to determine the pull's percentile deviation for each bottle reference and assess the impact of a spout change on the pull. It is worth noting that a basin

³BA Manufacturing eXperience is a data-centralizing platform that integrates data from various sources, including PSS via OPC. It offers real-time dashboards, visualizations, and recommendation tools to enhance plant operations [31].

change inevitably leads to a halt in the production line, making it a critical factor to consider in this evaluation.

Based on the analysis, it was observed that for AV2, a job change (change of the bottle reference) does not impact the furnace pull by more than 9%. Similarly, on a normal production day (without job or spout changes or malfunctions), the specific consumption should remain stable within a range of 2.5%. Moving on to AV4, the findings indicate that the pull deviation due to a job change is around 8%, while the specific consumption deviation remains below 2.3%.

Determining the values for AV5 proved to be more challenging since certain bottle references displayed a higher percentile deviation compared to other furnaces, approaching the magnitude of deviation seen during a spout change. As a result, the calculated percentile deviation for the pull on AV5 was 10.6%. However, it is worth noting that AV5 is the most efficient furnace in the Avintes facility, resulting in a slightly lower specific consumption deviation of 2.4% compared to AV2.

These conclusions provide insights into the impact of job changes and the overall stability of the furnaces in terms of pull and specific consumption, aiding in the identification of abnormal production days.

Subsequently, to prepare the dataset for model training, any values that were found to be below the threshold, as determined through the aforementioned process, were substituted with the mean value of the respective features. This data preprocessing step ensures the dataset is normalized and enables the models to learn from consistent and reliable input. By replacing values below the calculated threshold with the mean, there's a certainty that the trained models are not influenced by potentially erroneous or abnormal data points, enhancing the overall accuracy and robustness of the models during subsequent analysis and predictions.

Table 4.1: Percentage of filtered data for each furnace

Furnace	AV2	AV4	AV5
Percentage of filtered data	27.01%	27.78%	36.97%

After eliminating these irregularities and analyzing the graphs (Appendix A.2), it becomes evident that the consumption tends to follow a linear trend. Consequently, the ML tool developed in this study utilizes regression models such as linear, polynomial, Ridge, LASSO, and decision tree models.

4.2 Analysis and Forecasting for Energy Consumption Prediction using Data Log Values

The development of the ML model holds significant potential in optimizing furnace operations. Accurate consumption predictions can assist in proactive planning, resource allocation, and identifying areas for efficiency improvement.

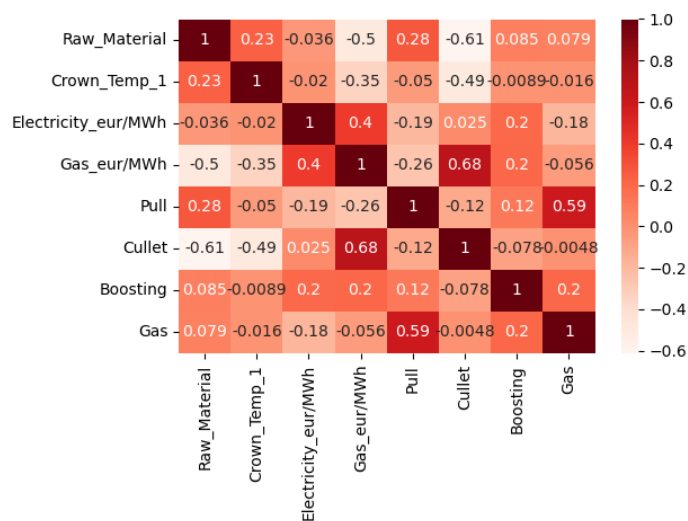
The DataFrame used for analysis contained values from 2022. It comprises labeled variables encompassing important aspects of the production process. The recorded features included the date, providing a chronological reference for the data. "Raw_Material", represents the amount of raw material utilized to form glass in the furnace. "Crown_Temp_1", measured in degrees Celsius, indicated the furnace's crown temperature. The expenses associated with electricity consumption, measured in euros per megawatt-hour (€/MWh), and gas consumption, also measured in €/MWh, were documented. The "Pull" feature was recorded in kilograms. The proportion of cullet in relation to the total quantity of cullet and raw material used was expressed as a percentage in the "Cullet" feature. The "Boosting" feature was measured in kWh for each furnace. The "Gas" feature denoted the amount of gas consumed by each furnace, expressed in units of Nm³. Lastly, the "Color_Code" feature was represented by a string of 2 characters (e.g., AS, UV, AM, etc).

4.2.1 Impact Analysis of Features on Furnace Consumption

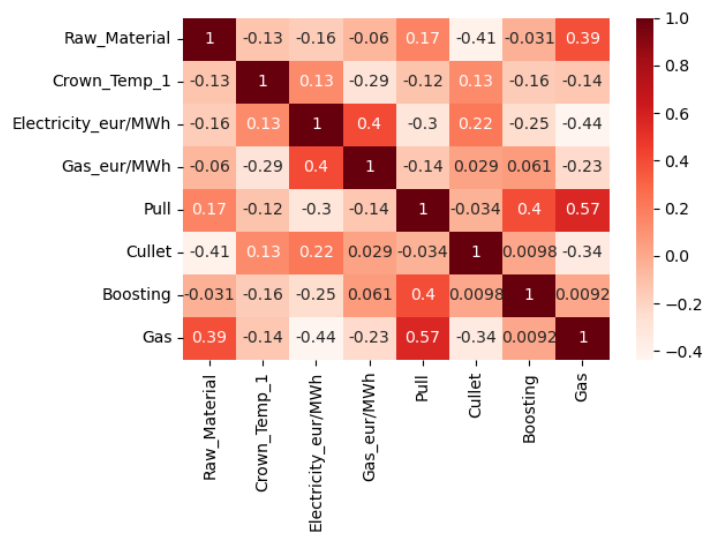
Leveraging the insights gained from the data analysis, an ML model was developed. This model was trained using the collected data to accurately predict furnace gas and electric consumption. The model aimed to provide accurate and reliable predictions for future furnace consumption by considering historical consumption data.

As discussed in Section 4.1, predicting energy consumption in the glass molting process requires careful consideration of various factors, including pull, cullet, glass color, energy prices, and furnace crown temperature.

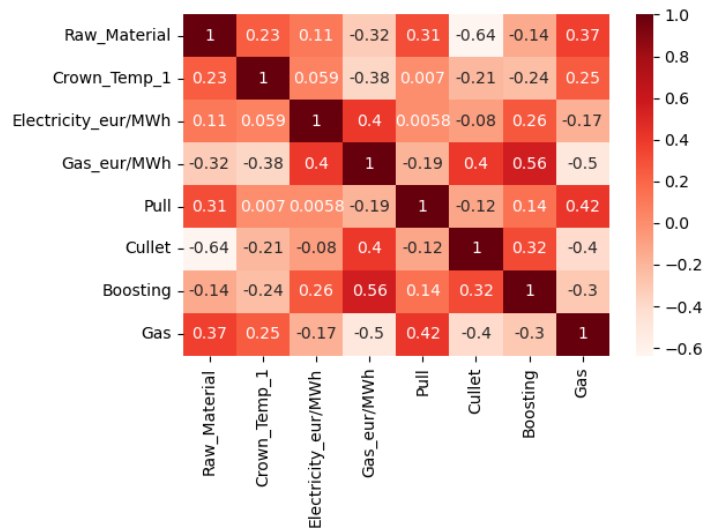
Upon analyzing the dataset, the initial step involves identifying the features that exhibit stronger correlations with the energy consumption of each furnace. This analysis allows us to determine which specific variables have a more significant impact on the consumption patterns of individual furnaces. By understanding these correlations, we can gain valuable insights into the factors that most significantly influence energy usage in the glass melting process.



(a) AV2



(b) AV4



(c) AV5

Figure 4.2: Feature correlations

To facilitate the understanding of the correlation between variables, the heat map function from the *seaborn* Python library was employed to represent the correlation values visually. The results are depicted in Figure 4.2.

Then, a plot was created to analyze the impact of glass color on total energy consumption (gas and electrical). The plot in Figure 4.3 depicts the total specific consumption for each color, visually representing how different glass colors affect overall furnace energy usage. Additionally, average total energy consumption values were calculated for each color, providing insights into the extent to which color influences consumption levels. By examining these results, shown in Table 4.2, we can better understand the relationship between glass color and energy usage.

After considering the data and variations in color since January 2022, it's clear that the influence of color on energy consumption is not significant enough in this time frame to be a significant factor in the developed model. The observed color variations mainly consist of different shades of the same color, which explains the minimal difference in energy consumption across these variations. Therefore, it may not be necessary to include color as a significant factor when analyzing and predicting energy consumption in this particular context.

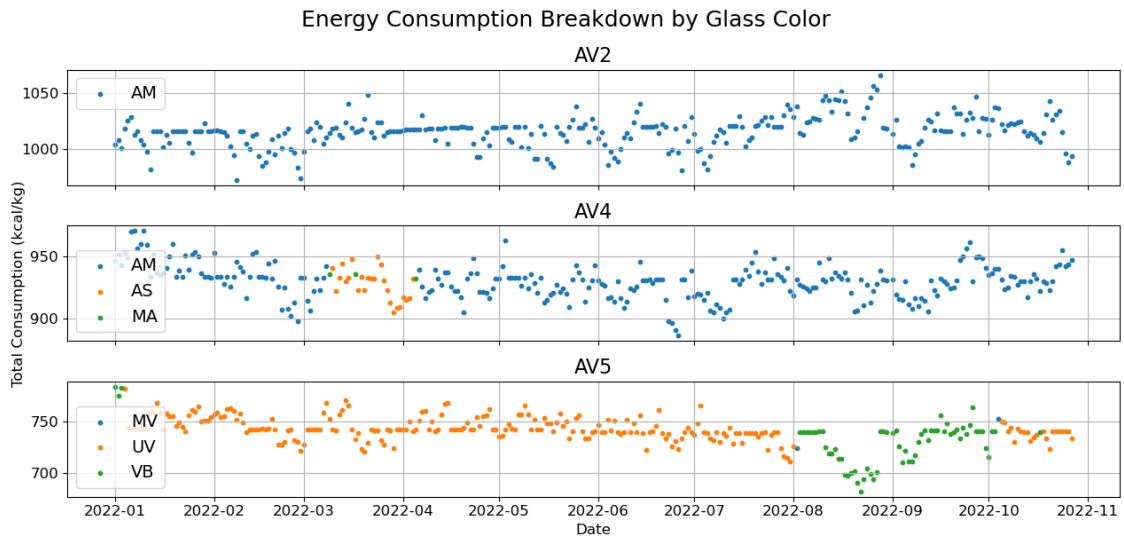


Figure 4.3: Total (gas and electrical) specific consumption of Avinte’s furnaces by color

Table 4.2: Average of total specific consumption of each furnace by color

Furnace	AV2	AV4	AV5
Color	(kcal/kg)	(kcal/kg)	(kcal/kg)
AM (amber)	1015.258	930.3996	————
AS (strong amber)	————	934.6722	————
MA (dark amber)	————	927.9933	————
MV (antique green)	————	————	738.7059
UV (UV green)	————	————	743.8408
VB (dark green)	————	————	730.4851
Percent difference	0.00%	0.717%	1.81%

After analyzing Figure 4.2, it becomes evident that, in general, there is a weak correlation between furnace consumption and the other features. However, it is worth noting that there is a moderate correlation between the pull and gas consumption features in all furnaces, which was already expected. This correlation is logical as a larger quantity of molten glass leads to higher energy consumption by the furnace. Hence, we can conclude that the "pull" feature significantly

impacts furnace energy consumption, whereas the remaining features exhibit weak correlations with consumption.

4.2.2 Development of Predictive Models for Furnace Consumption

The proposed ML tool functions as an advanced calculator with the objective of utilizing planned pull values for the upcoming week (data from "Tiragens.xlsx") to predict furnace consumption and assist in establishing consumption targets for the plant. The Python application generates outputs for each furnace's daily kWh and kcal/kg consumption throughout the week. Additionally, it provides a daily estimation of the expected expenses⁴ for gas and electricity.

An investigation was conducted to determine the best-performing model among a range of regression models, including linear regression, polynomial regression, ridge regression, LASSO regression, and decision tree regression. This selection was motivated by the assumption that the furnaces' consumption remains relatively constant.

To optimize the performance of the polynomial, ridge, and LASSO models, Python's *Grid-SearchCV* was employed. This approach allowed for identifying the best parameters for each model, enabling fine-tuning and improving their accuracy.

Various test scenarios were conducted to determine the optimal combination of training and testing data and to identify the most suitable model for the application. The methodology is further explained in Section 5.2. These scenarios were designed to evaluate the models' performance under different conditions and periods, from one week to one year.

4.3 Exploring an Approach for Furnace Consumption Optimization

The study of the optimization of the molting process focused on two key components: gas combustion and electric boosting. The aim was to determine the optimal combination of electric boosting and gas usage while considering the limitations of the furnace infrastructure.

In BA, the relationship between the pull and gas and electrical consumption was initially examined using Excel, which revealed an inverse correlation - as the pull increased, the specific consumption decreased. This correlation is logical, as a larger quantity of glass being molten would result in more efficient energy utilization.

To further explore the effectiveness of studying this relationship, linear, LASSO and ridge regressions were trained using Python's *scikit-learn (sklearn)* library. The linear, LASSO, and ridge regression models showed significant similarity. Table 4.3 compares the results of *sklearn*'s linear regression and the previously determined Excel regression that served as a reference for this study.

Upon examining the values in Table 4.3, it can be observed that the overall r^2 score is generally low, although still higher than the r^2 score obtained from the Excel regression. This outcome was expected since the data points tend to cluster within a specific range of pull, and consumption is

⁴To maintain confidentiality, the values employed herein are fictitious and do not represent actual data.

influenced by numerous other variables, making it challenging to establish a strong relationship based solely on pull.

Table 4.3: Comparison of Excel’s calculated regression and *sklearn*’s linear regression for analyzing gas and boosting in furnaces

			Excel	Linear
Gas	AV2	y =	-2.316 x + 1551.1	-2.258 x + 1535.1
		MSE	2187.8032	—————
		Training MSE	—————	1950.9961
		Testing MSE	—————	2717.8243
		r ²	0.25573461	0.20820936
	AV4	y =	-5.125 x + 1992.5	-5.019 x + 1964.3
		MSE	2222.0382	—————
		Training MSE	—————	2230.5084
		Testing MSE	—————	2202.9222
		r ²	0.74151225	0.74315502
	AV5	y =	-1.201 x + 1208.8	-0.801 x + 1035.4
		MSE	1277.3747	—————
		Training MSE	—————	999.4964
		Testing MSE	—————	1249.7466
		r ²	0.074561832	0.18222222
Boosting	AV2	y =	-0.653 x + 292.33	-0.354 x + 207.49
		MSE	185.8544	—————
		Training MSE	—————	171.8569
		Testing MSE	—————	170.0001
		r ²	0.065964827	0.21874942
	AV4	y =	-0.723 x + 321.39	-0.467 x + 266.56
		MSE	380.6266	—————
		Training MSE	—————	353.5631
		Testing MSE	—————	362.5357
		r ²	0.069499910	0.11231911
	AV5	y =	-0.240 x + 175.12	-0.0216 x + 80.571
		MSE	228.4196	—————
		Training MSE	—————	164.0673
		Testing MSE	—————	183.9678
		r ²	-0.33673025	0.0071880091

Similarly, evaluating the MSE allows us to assess the accuracy of the regression models in predicting specific consumption based on pull. A lower MSE indicates better model performance,

with more minor differences between predicted and actual values. Comparing the MSE demonstrates that, except for AV2's gas consumption, the linear regression performs well and generalizes effectively.

The r^2 score and MSE suggest challenges in establishing a strong relationship between pull and specific consumption. It implies that other variables besides pull significantly influence the variations in specific consumption.

In conclusion, the utilization of Python's *scikit-learn* library for determining linear regressions generally provided more accurate and reliable results. It could potentially be used as a viable alternative to Excel in future analyses within the company, enhancing the plant's decision-making process regarding energy consumption optimization. Only AV2's gas consumption *sklearn*'s regression did not yield superior results compared to the Excel regression. The graphs with the results are presented in Appendix A.3.

Next, the relationship between the percentage of boosting and the pull was explored, and no correlation was observed across the three furnaces, as shown in Figure 4.4.

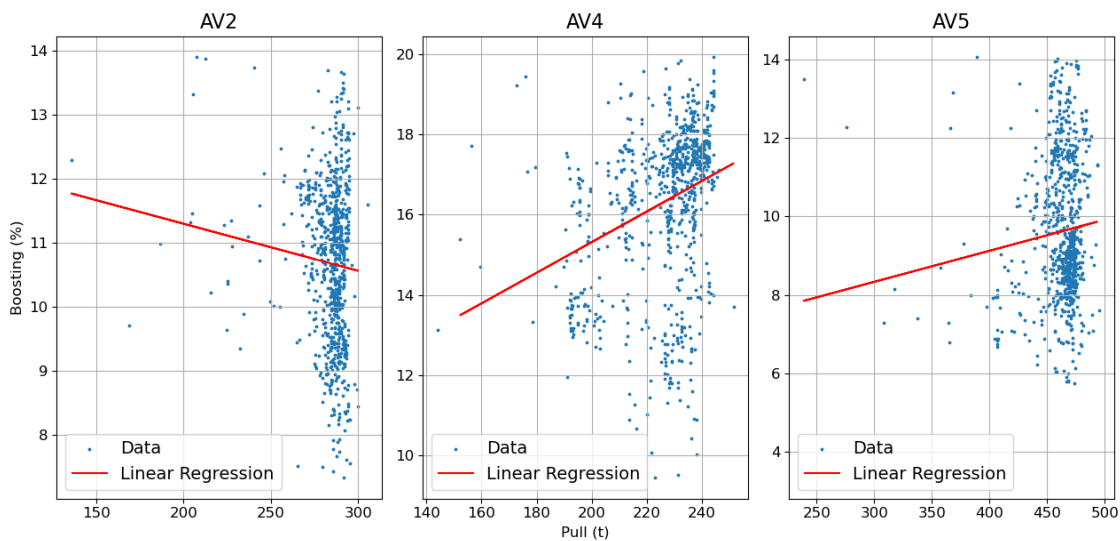


Figure 4.4: Boosting percentage vs. pull in tonnes

Given the understanding that furnace consumption is influenced by multiple variables beyond pull, an investigation was conducted to study the impact of fixing the percentage of cullet on the results. By focusing on this subset of data, we can examine any patterns or relationships that emerge between the variables of interest within this context.

Additionally, to explore the potential relationship between the temperature of the furnaces' crown and consumption, plots were generated to visualize the variations in temperature and boosting percentage over time. This analysis aimed to identify any significant patterns or relationships between these variables.

Figure 4.5 presents the plotted data specifically containing values with fixed cullet. By analyzing these plots, insights can be gained regarding the fluctuations of the temperature of the

furnaces' crown and the boosting percentage over time. This analysis also allows us to investigate whether any meaningful correlations exist between these variables. Additionally, it provides an opportunity to assess the impact of fixing the cullet percentage on the correlation between the boosting percentage and pull. By examining these relationships, we can further understand these variables' dynamics and potential influences on furnace performance.

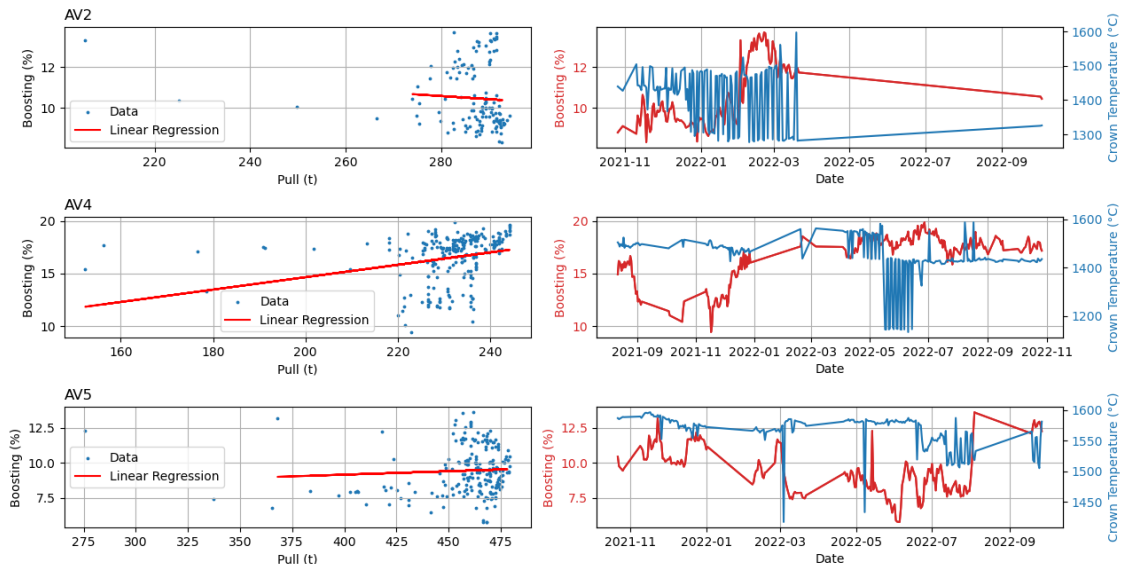


Figure 4.5: Analysis of boosting with cullet within a 5% range

However, it is important to note that establishing significant correlations was challenging due to the limited data range and insufficient temperature variability. Although there were instances where an increase in temperature corresponded to an increase in boosting percentage, and vice versa, no significant correlation was observed among the different furnaces. Despite these limitations, the analysis provides valuable insights into the relationship between the temperature of the furnaces' crown and the boosting percentage, indicating a potential association that warrants further investigation with a larger and more diverse dataset.

This analysis reinforces the previous findings, indicating that a combination of factors beyond a single variable influences furnace consumption. The results suggest that the correlation between furnace performance and the variables examined: gas and electrical consumption, boosting percentage, cullet percentage, pull, and crown temperature is complex and not easily determined. The limitations of the dataset, including its size and variability, may have contributed to the absence of strong correlations.

4.4 Estimating Consumption using Models Transferred from Other Furnaces

Transfer learning involves leveraging knowledge or labeled data from related fields to improve the performance of a ML algorithm in a target domain. It allows for the transfer of learning from one environment to another, enabling the development of accurate models even when specific training data is scarce or expensive to obtain.

To explore the application of transfer learning in glass-melting furnaces, an attempt was made to use the decision tree model trained with AV4's data to predict the consumption of furnace AV2. AV2 and AV4 are considered the most similar furnaces among the three in Avintes. However, an essential factor to consider is the age of the furnaces. To address this, an exclusion criterion based on furnace age was implemented. As AV2 and AV4 have a six-year difference, data from 2017 was utilized for AV2, while data from 2022 and 2023 was used for AV4. The available data for analysis was limited to the period from 01-12-2016.

Subsequently, the correlation between specific consumption (in kcal/kg) and the percentage of cullet used was computed for AV2, resulting in a value of -0.2593. To enhance the similarity between AV2 and AV4, AV2's cullet percentage values were adjusted by reducing them by 8%. Then, the consumption values were proportionally increased based on the correlation factor to make them more comparable to AV4.

This study performed a performance comparison of the method for one-month and one-year intervals. The results obtained are presented below in Table 4.4, while the corresponding graphs can be found in Appendix A.4.

Table 4.4: Comparison of performance metrics for AV2's model trained with AV2 and AV4 data, and vice-versa

		AV2 model		AV4 model	
		AV2 data	AV4 data	AV2 data	AV4 data
1 Year in 2017-2016	Training RMSE	3529.288	—	—	2562.637
	Testing RMSE	3900.804	11359.27	44642.84	3938.395
	r ²	0.785390	-2.33293	-25.1136	0.453375
1 Month	Training RMSE	1243.204	—	—	1833.829
	Testing RMSE	3005.316	30891.03	48033.82	2358.066
	r ²	0.744955	-21.1131	-43.4497	0.852173

When attempting to estimate the consumption of AV2 using the model calculated for AV4, and vice versa, it was found that the accuracy of the results was limited. The testing error was significantly large, and the r² score was found to be well below zero. Upon visual inspection of the graphs, it was observed that the prediction line deviated significantly from the actual furnace data, indicating poor performance and a lack of alignment between the prediction model and the furnace's consumption patterns.

Chapter 5

Validation and Evaluation

This chapter focuses on the testing and validation processes for the PSS data collection and the prediction model developed for the furnaces' energy consumption.

The first part of the chapter covers the testing and validation of the PSS data collection system. This involves physically visiting the flow meters on the plant floor and comparing the values with the ones in SCADA.

The second part of the chapter focuses on testing, validating, and evaluating the furnaces' energy consumption prediction model. This involves subjecting the model to extensive testing using historical data to assess its predictive capabilities from one week to one month and with various training data sizes. The model's performance is evaluated against known consumption values to measure its accuracy and precision.

The evaluation of the prediction model involves analyzing its strengths and limitations, assessing its robustness, and determining its suitability for practical application. This includes evaluating factors such as prediction accuracy and computational efficiency.

By thoroughly testing, validating, and evaluating both the data collection system and the prediction model, this chapter provides insights into the reliability and effectiveness of these components. This process ensures that the data collected is accurate and trustworthy and the prediction model can deliver reliable energy consumption forecasts for the furnaces.

5.1 *PowerStudio* Validation Methodology

A meticulous methodology was employed to validate the accuracy of the values displayed in *PowerStudio*, involving physically traversing the plant floor and visiting each flow meter. During this process, a thorough comparison was made between the values observed on the counters and those presented in the SCADA system. A corrective factor was introduced into the device configurations to ensure precise decimal representation. This step aimed to validate the seamless communication between the sensors and the input/output devices connected to SCADA.

The table used to register the values of the flowmeters is shown in Table 5.1.

Table 5.1: Gas flowmeters values registration table

Registo Diário de Contadores				Nome	
				Data	__/__/__
				Hora	__:__
	AV 2		AV 4		AV 5
Refiner		Refiner		Refiner	
F 20		F 41		F 51	
F 21		F 42		F 52	
F 22		F 43		F 54	
A 20		A 41		A 51	
A 21		A 42		A 52	
A 22		A 43		A 53	
E 21		E 41		E 51	
E 22		Forno Retração		E 52	
		FR 241			
		FR 242		E 53/54	
		FR 5			

Gás Total	
Fornos	
Outros	

PH	
Etari	
AV5	
AV2	

Osmose	
Av 2/4	
Av 5	

Decoração	
Água L43	
Ele.Q.N.	
Água Caldeira	
Gás Caldeira	
Arca Velha	
Arca Nova	

Norcasco	
Ele. PT5	
Gás	
Água	
Água Etari	

Água	
Companh	
Piscina	
Rio	
Poço	
Compress	

By implementing this methodology, we could effectively verify the consistency and reliability of the data displayed in *PowerStudio*, providing a solid foundation for further analysis and decision-making.

5.2 Furnace Consumption Prediction Tool Testing Methodology

Two test scenarios were executed in this Section. The first scenario aimed to determine which regression models outperformed the others, while the second scenario shifted the focus to determining the optimal combination of training and prediction periods to identify the combination that yielded the most accurate energy consumption predictions.

These tests involve evaluating various regression models and the performance of each combination of training and prediction periods using appropriate metrics: r^2 score, MAE, training RMSE, testing RMSE, and Standard Deviation (SD).

The regression models were obtained and analyzed with historical data from the "*Registos consumos_auto_1.xlsx*" file, which contains all the necessary data. To ensure the reliability and consistency of the data, information before 2022 was excluded from the study. This decision was made because the file provides more reliable data from 2021, and any inconsistencies resulting from maintenance work on the furnaces in 2021 were resolved by 2022.

A DataFrame was subsequently generated using the data from the file mentioned earlier, incorporating the following columns: date, pull (kg), boosting (kWh), gas consumption (Nm³), and PCI (kWh/m³). Additionally, the total consumption absolute values were calculated from these features to serve as the target variable for the regression models. The objective of using the total absolute consumption as the target variable was to assess the algorithm's ability to detect significant fluctuations. This choice was made because the specific consumption tends to remain more constant than the absolute consumption.

As previously mentioned, the models were evaluated using r^2 score, MAE, training RMSE, testing RMSE, and SD. Using these metrics gives a comprehensive understanding of the regression model's performance. The r^2 summarizes how well the model fits the data. MAE provides an average magnitude of the prediction errors. The training and testing RMSE evaluate the average magnitude of the prediction errors, and the comparison between them indicates whether the model is overfitting or underfitting. SD gives insights into the variability of the predictions.

RMSE directly quantifies the average magnitude of prediction errors, while SD measures the dispersion of the actual values. Although RMSE and SD reflect different aspects of variability, they are related in that a smaller RMSE suggests a reduction in the spread of prediction errors, which is associated with a smaller SD.

Overall, it is essential to consider multiple metrics to assess the model's strengths and weaknesses thoroughly.

5.2.1 First testing scenario

An investigation was conducted using linear, polynomial, ridge, LASSO, and decision tree regressions to determine the better-performing model. The regression models were tested six times for the following periods:

1. One-month testing period:
 - Period 1: From 01-01-2022 to 01-02-2022
 - Period 2: From 01-04-2023 to 01-05-2023
2. Six-month testing period:
 - Period 1: From 01-01-2022 to 01-07-2022
 - Period 2: From 01-11-2023 to 01-05-2023
3. One-year testing period:
 - Period 1: From 01-01-2022 to 01-01-2023
 - Period 2: From 01-05-2022 to 01-05-2023

By testing the models on different periods, ranging from one month to one year, the performance and generalization capabilities of the models across various time horizons were assessed.

This approach provided insights into how well the different regression models performed on short-term and long-term predictions, enabling a comprehensive evaluation of their effectiveness.

The dataset was divided into training and testing sets using the `train_test_split()` function from the `sklearn` library. The testing dataset comprised 20% of the overall DataFrame, while the remaining 80% was used for training the regression models. This split ensured the models were evaluated on unseen data during testing. Code Section 5.1 shows an illustration of the code.

```

1 # # Prepare data to train the models
2 # Convert dates to Unix timestamps
3 filtered_data["Timestamp"] = filtered_data["Date"].astype("int64") / 10**9
4
5 # Split data into input and output variables
6 X = filtered_data[["Timestamp", "Pull"]]
7 y = filtered_data[["Total_kWh"]]
8
9 # Split the data into training and testing sets
10 X_train, X_test, y_train, y_test = train_test_split(
11     X, y, test_size=0.2, random_state=42)

```

Code Section 5.1: Python code for splitting the DataFrame in training and testing datasets

In these tests, the independent variables chosen from the DataFrame's features were "Pull" and "Timestamp." The selection of pull as a variable was based on its highest correlation with consumption, as identified in Subsection 4.2.1. Additionally, the performance of furnaces is influenced by their age, hence the inclusion of the timestamp feature. Considering that one of the tests spanned one year, the timestamp variable becomes significant to take furnace efficiency over time into account.

The models were evaluated using the previously mentioned metrics: r^2 score, MAE, training RMSE, testing RMSE, and SD. Code Section 5.2 demonstrates how these metrics were collected.

```

1 # # # # # Linear
2 model_linear = LinearRegression()
3 model_linear.fit(X_train, y_train)
4 ypred_linear_train = model_linear.predict(X_train)
5 ypred_linear_test = model_linear.predict(X_test)
6 # Display results
7 print("Data standard deviation: %.6f" % np.std(y_test))
8 print("\nR-squared score")
9 print(r2_score(y_test, ypred_linear_test))
10 print("\nMean absolute error")
11 print(mean_absolute_error(y_test, ypred_linear_test))
12 print("\nTraining mean squared error")
13 print(mean_squared_error(y_train, ypred_linear_train))
14 print("\nTesting mean squared error")
15 print(mean_squared_error(y_test, ypred_linear_test))

```

Code Section 5.2: Python code for determining metrics

To determine the optimal λ value for ridge and LASSO regression, a *GridSearchCV* approach was employed. The grid search was performed using ten values ranging from 0.001 to 1000. This technique systematically evaluated the performance of ridge and LASSO regression models across the range of λ values using negative MSE as the scoring metric to identify the best fit.

In the case of polynomial regression, a pipeline was utilized to streamline the process. The pipeline combined *PolynomialFeatures* and *LinearRegression*, enabling the creation of polynomial features and the subsequent fitting of the regression model. The hyperparameters considered in the grid search were the degree of the polynomial, ranging from 2 to 11, and the *fit_intercept* parameter, which determined whether an intercept term was included in the regression equation. Both True and False options were explored during the grid search to thoroughly assess the impact of the *fit_intercept* parameter on the model's performance. Negative MSE was the scoring parameter used for this regression model as well.

Determining a decision tree's maximum depth value is critical for balancing underfitting and overfitting. To find the optimal *max_depth* parameter, several decision trees were trained with values ranging from two to seven. The performance of each tree was evaluated using accuracy as the metric. As the depth of the trees increased, the training performance improved. However, beyond a certain point, the validation performance reached its peak and then started to decline. It was observed that the maximum depth of four yielded the highest validation performance, indicating the best trade-off between model complexity and generalization. This depth ensured that the decision tree achieved an appropriate level of accuracy without overfitting the training data, striking a balance between capturing intricate patterns and maintaining generalizability.

5.2.2 Second testing scenario

The objective of this test scenario was to identify the optimal combination of prediction and training periods using the best-performing models from the previous scenario. This study used polynomial and decision tree regressions for furnaces AV2 and AV4, and for furnace AV5, the same models were tested with the inclusion of the LASSO regression model.

The best-performing models were tested using an iterative process. Each regression model in this test scenario predicted energy consumption for a predetermined period. The models were trained using data from the chosen precedent training period, and predictions were made iteratively, composing a predefined prediction duration set to six months or one year. Code Section 5.3 shows the iteration performed for each user-specified combination. The prediction period consisted of intervals of one week, two weeks, one month, and two months, and the training period options included one, two, and three months. This resulted in twenty-four combinations of prediction and training periods that were tested.

```
1 PRED_PERIOD = 60
2 TRAIN_PERIOD = 90
3 PRED_DURATION = 365
4
5 # Define the specified date
```

```

6 specified_date = datetime.date(2023, 5, 1)
7 # Calculate the start and end dates for prediction
8 prediction_start_date = specified_date - datetime.timedelta(days=PRED_DURATION)
9 prediction_end_date = specified_date + datetime.timedelta(days=1)
10 (...)
11 # # Perform the iterative prediction process
12 while prediction_start_date <= prediction_end_date:
13     # Calculate the start and end dates for training
14     training_start_date = prediction_start_date - pd.DateOffset(days=TRAIN_PERIOD)
15     training_end_date = prediction_start_date
16
17     # Filter the training data for the current prediction week
18     training_data = df[(df['Date'] >= training_start_date) & (df['Date'] <=
19         training_end_date)]
20
21     # Split the data into input and output variables
22     X_train = training_data[["Pull"]]
23     y_train = training_data[["Total_kWh"]]
24     (...)
25     # Increment the prediction week
26     prediction_start_date += pd.DateOffset(PRED_PERIOD+1)

```

Code Section 5.3: Iteration in Python for obtaining results with a user-specified prediction and training period combination

This method aimed to capture short-term patterns and fluctuations in energy consumption while maintaining sufficient training data to enable accurate predictions for an extended duration. As a result, using the timestamp as an independent variable was no longer necessary. Instead, the model was constructed solely based on the "pull" variable to predict the absolute consumption values. By focusing on the pull variable, the model aimed to capture the key factor influencing energy consumption and generate reliable predictions without needing timestamp information.

To calculate the r^2 score, MAE, training RMSE, testing RMSE, and SD, two DataFrames were created: one to store the training data and another to store the predictions. Code Section 5.4 demonstrates how these metrics were calculated.

```

1 # Prepare an empty DataFrame to store the training and predictions values
2 training_df = pd.DataFrame(columns=['Date', 'y_train', 'Prediction'])
3 predictions_df = pd.DataFrame(columns=['Date', 'Prediction'])
4 (...)
5 # # Perform the iterative prediction process
6 while prediction_start_date <= prediction_end_date:
7     (...)
8     # Make the prediction
9     prediction = model.predict(prediction_data)
10
11     # Store the prediction in the predictions DataFrame

```

```

12     prediction_row = pd.DataFrame({'Date': pred_dates.squeeze(), 'Prediction':
13     prediction.flatten()})
14     predictions_df = pd.concat([predictions_df, prediction_row], ignore_index=True)
15     # Store the training data in the training DataFrame
16     training_row = pd.DataFrame({'Date': train_dates.squeeze(), 'y_train': y_train.
17     values.flatten(), 'Prediction': (model.predict(X_train)).flatten()})
18     training_df = pd.concat([training_df, training_row], ignore_index=True)
19     training_df = training_df.drop_duplicates()
20     (...)
21     pred_date_range = pd.date_range(predictions_df['Date'].values[0], predictions_df['
22     Date'].values[-1])
23     train_date_range = pd.date_range(training_df['Date'].values[0], training_df['Date'
24     ].values[-1])
25     y_true = df[df['Date'].isin(pred_date_range)]['Total_kWh']
26
27     # Calculate the R-squared score
28     r2 = r2_score(y_true, predictions_df['Prediction'])
29     # Calculate root mean absolute error (MAE)
30     mae = mean_absolute_error(y_true, predictions_df['Prediction'].values)
31     # Calculate root mean square error (RMSE)
32     train_mse = mean_squared_error(training_df['y_train'].values, training_df['
33     Prediction'].values)
34     train_rmse = np.sqrt(train_mse)
35     test_mse = mean_squared_error(y_true, predictions_df['Prediction'].values)
36     test_rmse = np.sqrt(test_mse)
37     # Calculate standard deviation (SD)
38     sd = np.std(y_true)

```

Code Section 5.4: Python code for determining metrics for the second test

In the initial trial, the metrics indicated a decline in the performance of the regression models compared to the previous test. The approach employed utilized the bagging technique to address this issue and enhance the results of at least one model. As mentioned in Subsection 2.2.1.2, bagging involves training multiple trees on resampled versions of the training data and averaging their predictions, which aims to mitigate the observed performance challenges. The *BaggingRegressor* class from Python's *sklearn* library was utilized to implement this technique. This class provided the necessary functionality to train an ensemble of regression trees on resampled training data and aggregate their predictions for improved performance. To ensure the reproducibility of the results, *random_state* was set to forty-two. Overall, there was a slight improvement in the performance of the decision tree model after applying the bagging technique.

5.3 Results and evaluation

Table B.3 presents the results from the first test scenario, and Appendix A.5 shows the corresponding graphs. Based on the metrics obtained, it is evident that both polynomial and decision tree regressions generally yield the best results. Although the MAE and RMSE values are not ideal, they

can be deemed acceptable considering the high magnitude of consumption values. However, the r^2 values are somewhat inconsistent. While some tests resulted in negative r^2 scores, others exhibited high values, reaching around 0.8. Notably, AV4 proved to be the furnace that was relatively easier to predict in terms of consumption and presented the most consistent values.

Tables B.4, B.5, and B.6 present the results from the second test conducted on furnaces AV2, AV4, and AV5, respectively. It was observed that, in general, the combination of a two-week prediction period and a three-month training period yielded the best results for all three furnaces. However, there were some variations in the optimal prediction duration. For AV4, the results were better when predicting consumption over six months. On the other hand, for AV2 and AV5, the results improved when the prediction spanned over a one-year duration.

Furthermore, the values for AV2 and AV5 raise concerns regarding their consistency. The SD should ideally be relatively similar for the polynomial and decision tree models since the metric depends on the original data, but the SD varies significantly. However, in the case of AV5, which was also tested with the LASSO model, it is noteworthy that SD results are similar to the SD results of the polynomial regression. This seems to indicate that the problem may lie in the configuration of the decision tree regression model. Additionally, it would be expected that the SD values would be larger for one year compared to six months, as the data range is double the size, i.e., there is a greater variability of values over a longer time span.

In contrast, AV4 demonstrates results that are logical and coherent. Considering that the same code was used for all three furnaces, it is plausible that the issue lies within the data itself. A solution could lie in reviewing the data used for AV2 and AV5 to identify any potential errors, inconsistencies, or outliers that might be contributing to the peculiar results.

In conclusion, based on the results obtained, the decision tree model with a bagging regressor performs best. The optimal combination of prediction and training periods is found to be two weeks for the prediction period and three months for the training period. This configuration yields the most accurate and reliable predictions for the given scenario. This model and parameter combination will be implemented in the ML prediction tool that was proposed at the start of this dissertation for utilization at the Avintes plant.

Chapter 6

Conclusions and Future Work

This chapter represents the culmination of this work, presenting the final conclusions derived from the research questions delineated in Section 1.4 and delving deeper into the results from the tests presented in Section 5.2, which were designed to address these research questions.

Chapter 6 also includes insights into the work developed in Avintes's SCADA system and the final application proposed at the beginning of this dissertation. It explores how these contributions have enhanced BA's energy monitoring and management.

The final part of this chapter contains a discussion on future work regarding this dissertation's subject and highlights the limitations encountered during its development. It explores potential directions for further investigation that could have been pursued given more time. Moreover, the limitations faced during the study are carefully examined to provide a comprehensive understanding of the research's scope and the constraints that were encountered.

6.1 Discussion

Based on the results presented in Section 5.3, the furnace AV4 demonstrated the most promising outcomes. Therefore, further exploration of the insights gained from the conducted tests will primarily focus on AV4.

1. Which ML regression model is the most accurate in predicting furnace consumption in the glass manufacturing industry?

To address the first research question, which examines the most accurate ML regression model for predicting furnace consumption in the glass manufacturing industry, the results of the first test scenario indicate that both polynomial regression combined with *GridSearchCV* and decision tree combined with a bagging regressor yield the best outcomes. However, it is worth noting that the decision tree model tends to overfit the data more than polynomial regression.

Answering the first research question suggests that both models have their advantages and disadvantages, and the selection should be based on the specific requirements of the study. Linear

regression models, in general, may not be suitable for this application. Tree-based models exhibit significant potential, and polynomial regression works best in conjunction with *GridSearchCV* to avoid relying on a single value for the degree of polynomial regression. Based on the analysis of the results, the decision tree model appears to perform better for longer prediction periods, while polynomial regression yields better results for shorter prediction times. The decision tree model demonstrates its strength when making predictions over extended periods. It captures complex patterns and relationships in the data, making it particularly effective for longer-term predictions.

2. What is the optimal combination of training and testing data sizes for predicting consumption?

Based on the insights derived from the second test scenario, it becomes apparent that the models generally exhibit superior performance when trained on a larger dataset and applied to predict a shorter period. Specifically, the results indicate that the model's performance is significantly compromised if the prediction period is longer than the training time. However, it is crucial to emphasize that the conventional practice of using a 70% training and 30% testing data split still holds importance. In developing this prediction tool, it is vital to consider the quantity and quality of the data collected. Striking the right balance between data quantity and data quality is crucial.

In other words, when making a one-week prediction, it is more beneficial to possess a set of consistently accurate historical data covering one month rather than having three months' worth of inconsistent data. This highlights the significance of reliable and consistent data for achieving better predictive performance. Striving for a robust and accurate dataset will substantially impact the model's performance more than merely increasing the data size.

Moreover, the second test also sheds light on the computational efficiency of the regression models. The decision tree model demonstrates superior processing time compared to polynomial regression. Despite a slight compromise in accuracy, the decision tree model completes the two-week prediction period and three-month training period for a one-year prediction span in only 25 seconds, while polynomial regression requires 106 seconds. As a result, the decision tree model was chosen for developing the final prediction tool.

The second test provides valuable insights for answering the research question. It highlights the importance of having sufficient historical data to train the model effectively. Additionally, it suggests that a shorter prediction period allows the model to make more accurate predictions. Considering the computational efficiency, the decision tree model outperforms polynomial regression. This factor plays a crucial role in selecting the final prediction tool, as it ensures faster processing time while maintaining acceptable accuracy levels.

In conclusion, the insights from the second test scenario indicate that a larger training dataset and a shorter prediction period enhance model performance. Furthermore, the decision tree model, with a two-week prediction period and three-month training data, is selected as the final prediction tool due to its superior computational efficiency and satisfactory accuracy.

6.1.1 Work Contributions

The goal defined for this dissertation was to assist BA Avintes in its energy monitoring and management objectives. The dissertation's focus on improving the SCADA system and maximizing its potential contributes significantly to BA Avintes.

Throughout the study, notable improvements have been made to Avintes's SCADA. It has become more user-friendly, offering options for selecting preferred languages. Additionally, the system now features automatic generation of reports on gas and solar panel energy consumption. Finally, adding alarms to detect flow meter malfunctions significantly improves energy monitoring within the plant. These enhancements to the SCADA allow for more streamlined energy monitoring and management processes.

Moreover, the development of the prediction tool holds great value for energy management at BA Avintes. This tool enables the assessment of projected energy consumption for the upcoming week based on the planned glass melting operations. By having insight into the expected energy requirements, BA Avintes can proactively manage energy resources and make informed decisions to optimize energy consumption. The results obtained from this prediction tool can be found in Table B.7 and Figure 6.1, while the corresponding code developed for the tool is provided in Appendix C.2.

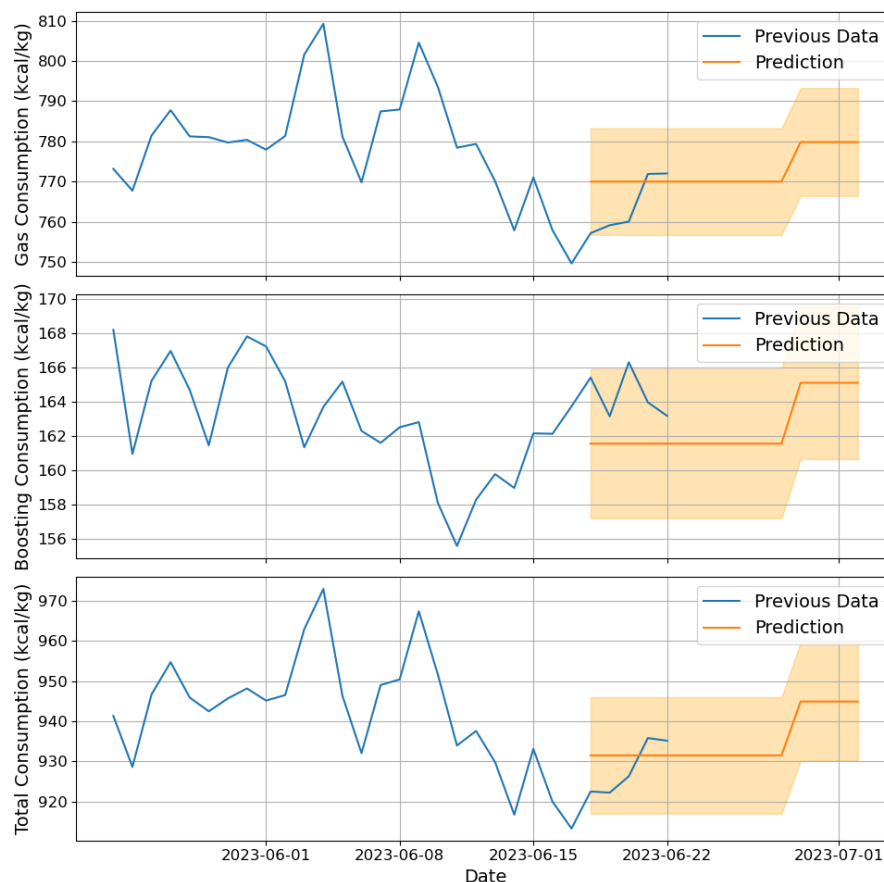


Figure 6.1: Graph result from running AV4's energy consumption prediction tool

Overall, the dissertation's contribution to Avintes's energy monitoring and management objectives is evident. The improvements to the SCADA system and the development of the prediction tool provide valuable tools and insights for efficient energy management practices, empowering BA Avintes to make data-driven decisions, enhance energy efficiency, and effectively achieve its energy management goals.

6.1.2 Limitations

One of the limitations encountered in the study was the quality of the data. The data contained numerous inaccuracies and inconsistencies, which posed a significant challenge when filtering and preprocessing it for analysis. Dealing with such data issues can be time-consuming and affect the results' reliability and accuracy.

Another limitation was the lack of strong correlation between certain variables, despite their practical interdependence. While it is known that variables such as gas and electrical consumption, pull, furnaces' crown temperatures, cullet, quality of bottles produced, and glass color are interconnected in practice, the available data did not exhibit enough variability or correlation to study these relationships comprehensively. This lack of variability is due to operational constraints within the factory, as extreme conditions (e.g., very high or very low temperatures, high cullet percentage) would be structurally unsafe for the furnace or result in poor product quality. Consequently, data capturing such extreme conditions is limited or absent.

As a result, the prediction tool developed for this study primarily considered a single variable, which may not fully capture the complex interactions and dependencies between the different factors influencing energy consumption. While this approach is pragmatic given the limitations of the available data, it is acknowledged that a more comprehensive understanding of the system's dynamics would require considering additional variables.

In summary, the limitations related to data quality, inconsistencies, and the lack of variability in certain interconnected variables constrained the depth of analysis and the ability to study complex relationships within the glass manufacturing process. Despite these limitations, the study aimed to provide useful insights and develop a prediction tool that could aid in energy management to a reasonable extent based on the available data.

Another limitation encountered during the study was the lack of timely technical support from *Circutor*, the company responsible for providing support for the SCADA system or related components. This limitation hindered the resolution of certain issues that arose during the implementation or configuration of the SCADA system.

However, despite this limitation, alternative solutions or workarounds were implemented to mitigate the impact. For example, in the case of the alarm system, although the desired SMS notification functionality was not operational, an interim solution was implemented with the use of an alarm light to alert operators to activated alarms. Such adaptations were made to ensure that the study could progress and achieve its objectives to the best possible extent, considering the limitations faced.

6.2 Future Work

If this study had more time, some additional improvements and integrations could have been implemented. One such enhancement would have been establishing a connection between the prediction tool and the SCADA system, enabling the generation of alarms when the plant's operations consumed more energy than predicted. The proposed approach for this integration involved utilizing OPC Router and MQTT¹. However, due to the complexity of the connection, it was not feasible to implement within the study's timeframe.

Furthermore, further improvements to the SCADA system were desired. Although progress had been made, particularly in managing variables and the GUI for natural gas, some areas still required attention. The water-related screens within the SCADA system are inaccurate in reflecting the system in place at the plant and need further refinement. An issue with the alarms not triggering SMS notifications persisted despite numerous tests. To rectify this, technical support from *Circuitor* is deemed necessary. In the interim, a solution was implemented wherein an alarm light would turn on when an alarm was activated. This allowed operators to be aware of the alarm and refer to the SCADA application to identify the specific alarm and address the issue accordingly.

Given additional time, these improvements and integrations could have been pursued, further enhancing the functionality and usability of the SCADA system and facilitating better energy monitoring practices at BA Avintes.

¹Message Queuing Telemetry Transport is a remarkably straightforward and lightweight messaging protocol that facilitates subscription and publication of messages.

Appendix A

Images

A.1 GUI Improvements (3.1.2)


 Gas meters reading		Month / Year 05/2023
Natural Gas		
Meter	Readings	Factor
Lehr 20	8513.4	x 1
Lehr 21	5065.2	x 1
Lehr 22	7069.5	x 1
Feeder 20	7830.2	x 1
Feeder 21	8059.3	x 1
Feeder 22	2351	x 1
Refiner AV2	34929	x 1
Mold Lehr L20/21	637.82	x 1
Mold Lehr L22	629.06	x 1
Lehr 41	5292.5	x 1
Lehr 42	4509.1	x 1
Lehr 43	9363.6	x 1
Feeder 41	19730.4	x 1
Feeder 42	19207.3	x 1
Feeder 43	20542.3	x 1
Refiner AV4	63059	x 1
Mold Lehr AV4	702.05	x 1
Lehr 51	8226.3	x 1
Lehr 52	6311.6	x 1
Lehr 53	10897.6	x 1
Lehr 54	9201.5	x 1
Feeder 51	51954	x 1
Feeder 52	23892	x 1
Feeder 53	29362	x 1
Feeder 54	33382	x 1
Refiner AV5	53762	x 1
Mold Lehr L51	603.03	x 1
Mold Lehr L52	928.42	x 1
Mold Lehr L53/54	914.54	x 1
Cullet treatment	18162.7	x 1
Decoration line lehr	12904.7	x 1
Decoration line lehr (new)	1077.42	x 1
Boiler - DEC - Sleeves	3976.6	x 1
Retraction furnace 24.1	3194.9	x 1
Retraction furnace 24.2	1353.8	x 1
Retraction furnaces AV5	3434.6	x 1
Propane Gas		
Boiler - propane air (m3)	0	x 1
The responsible person		Date
		29/06/2023

Figure A.1: Automatic gas report

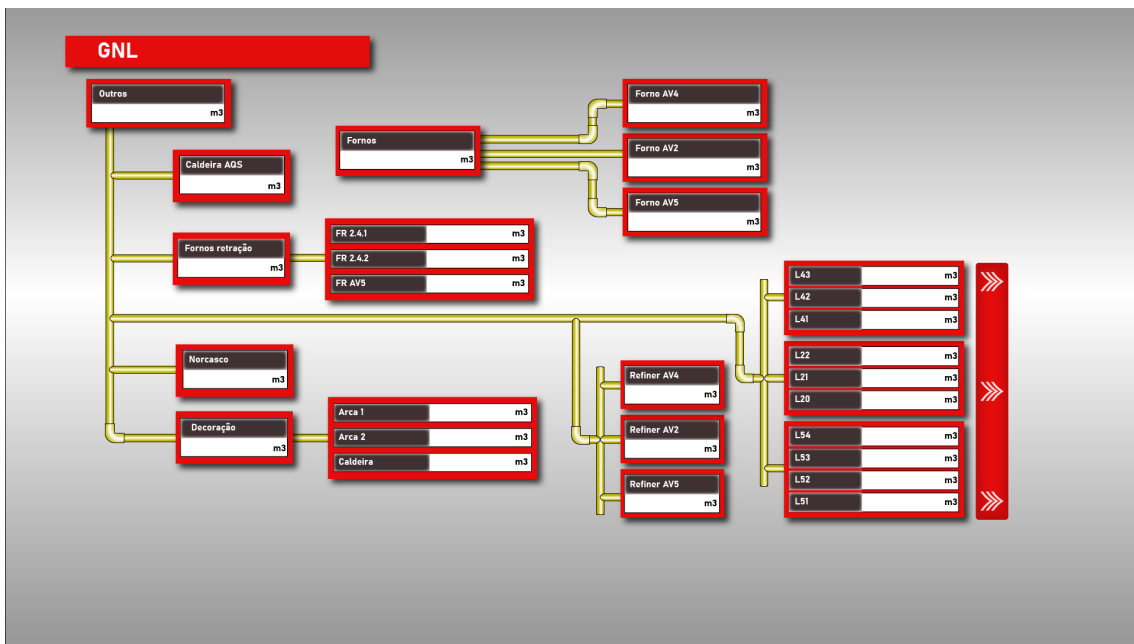


(a) Before

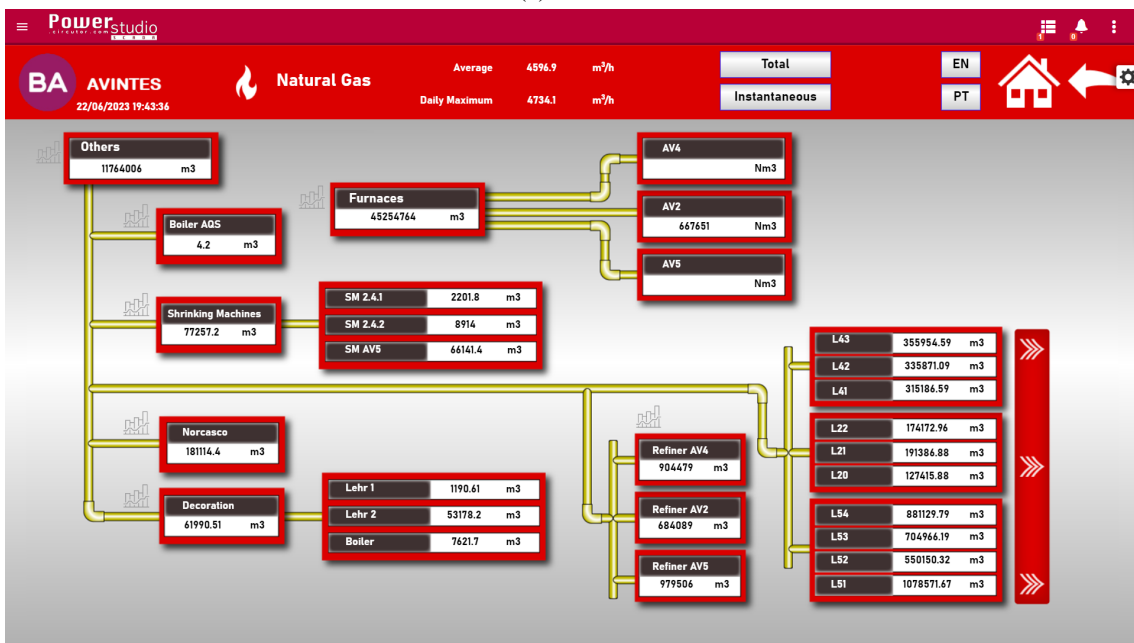


(b) After

Figure A.2: Main screen

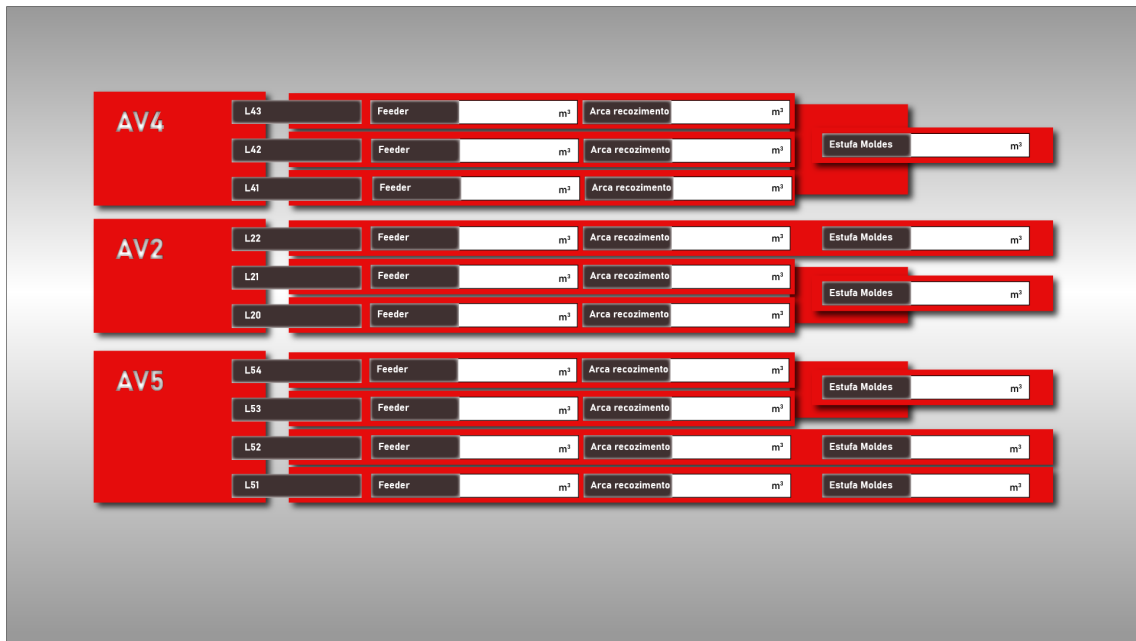


(a) Before

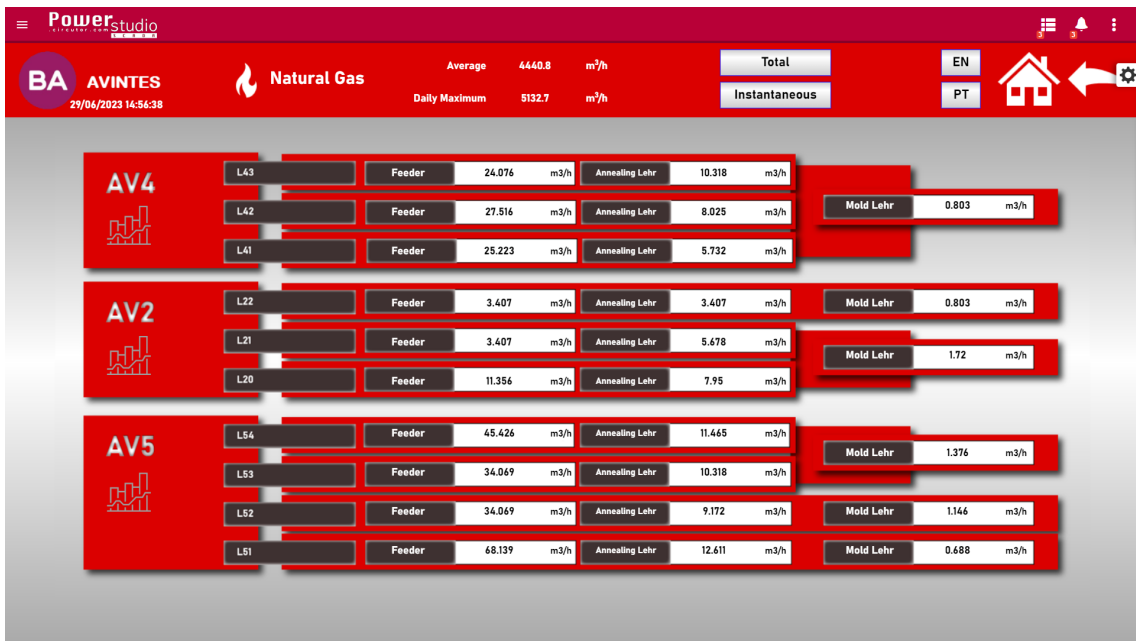


(b) After

Figure A.3: Plant's gas consumption screen

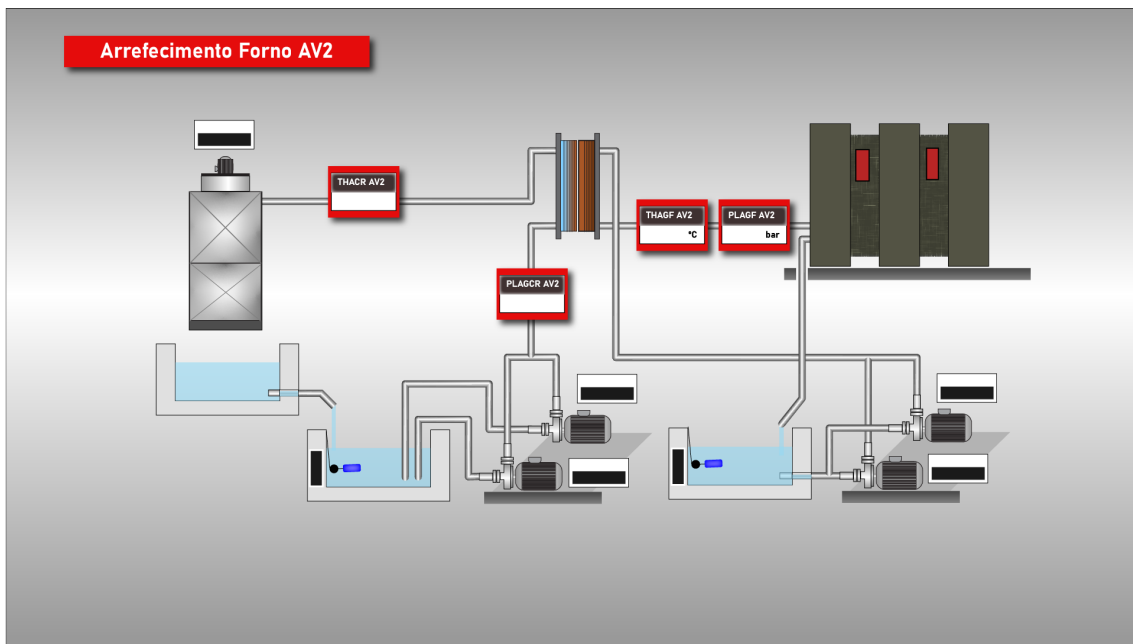


(a) Before

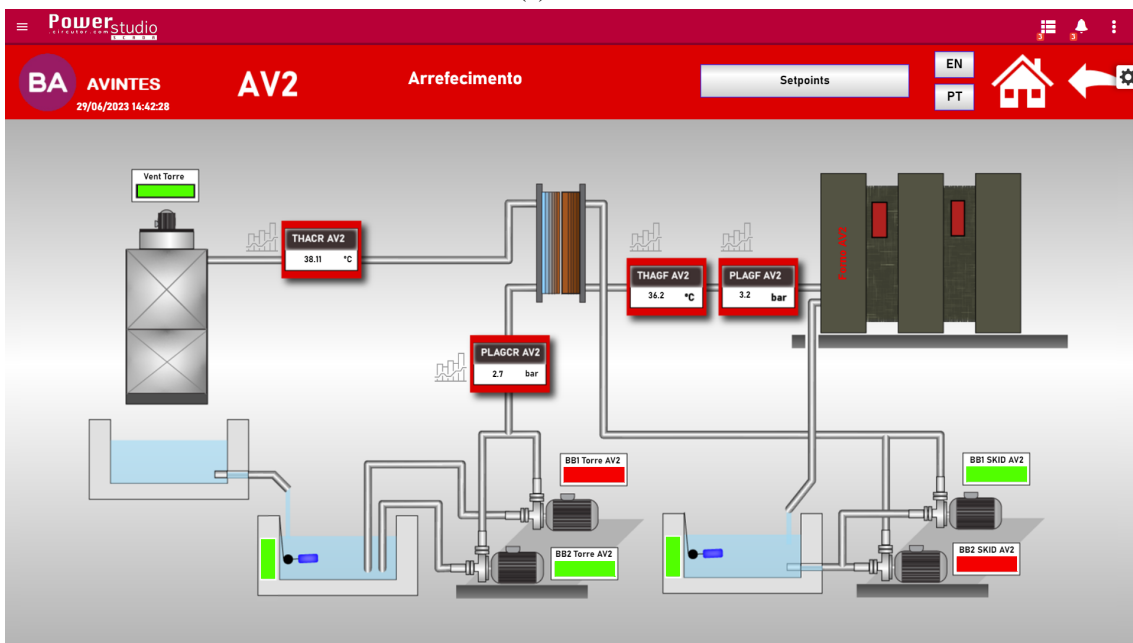


(b) After

Figure A.4: Production lines' gas consumption screen



(a) Before



(b) After

Figure A.5: Furnace AV2 cooling system screen

A.2 Collected Data vs. Filtered Data

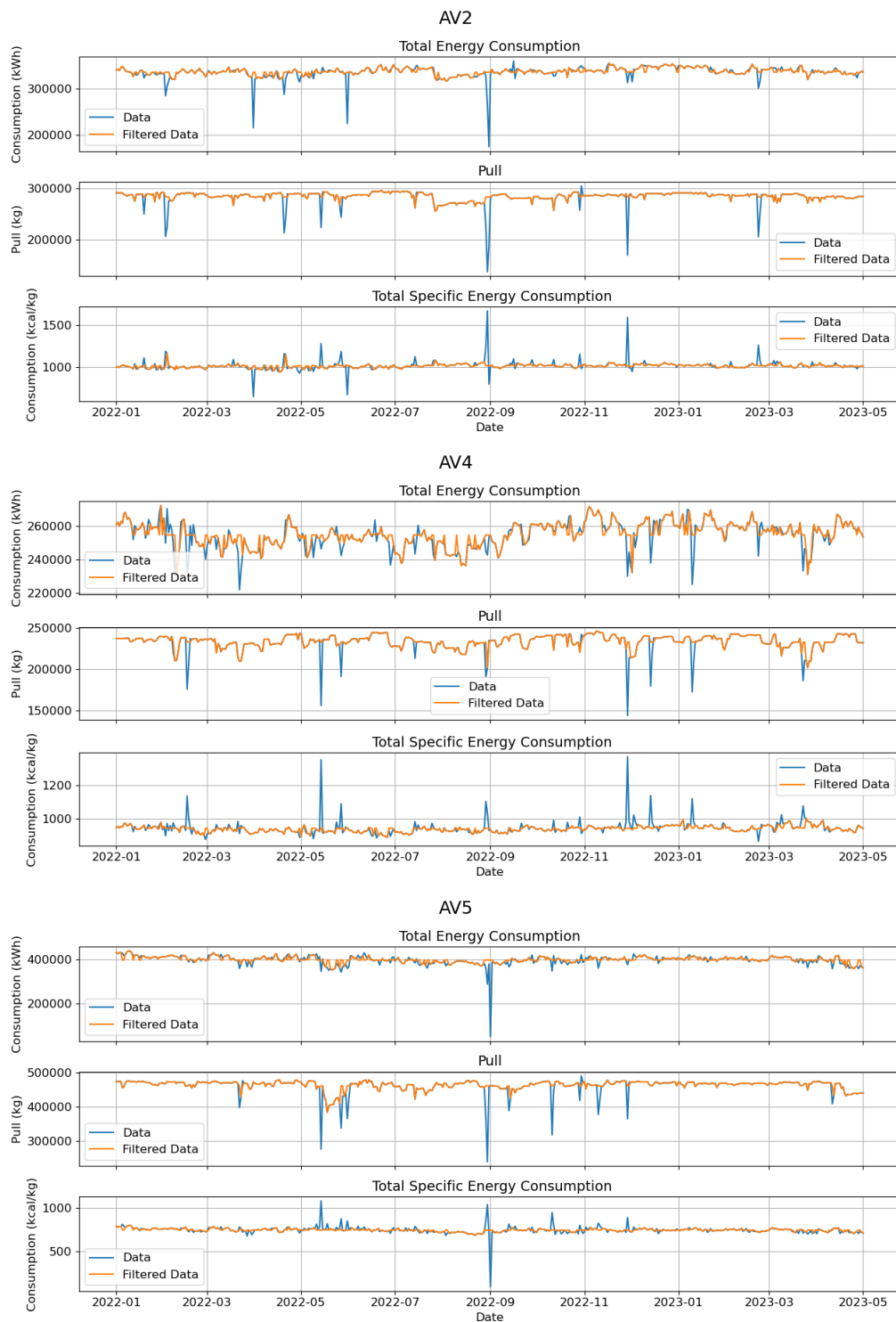


Figure A.6: Comparison of collected and filtered total consumption, pull and specific consumption data (4.1)

A.3 Comparison of Regression Models for Predicting Specific Consumption based on Pull

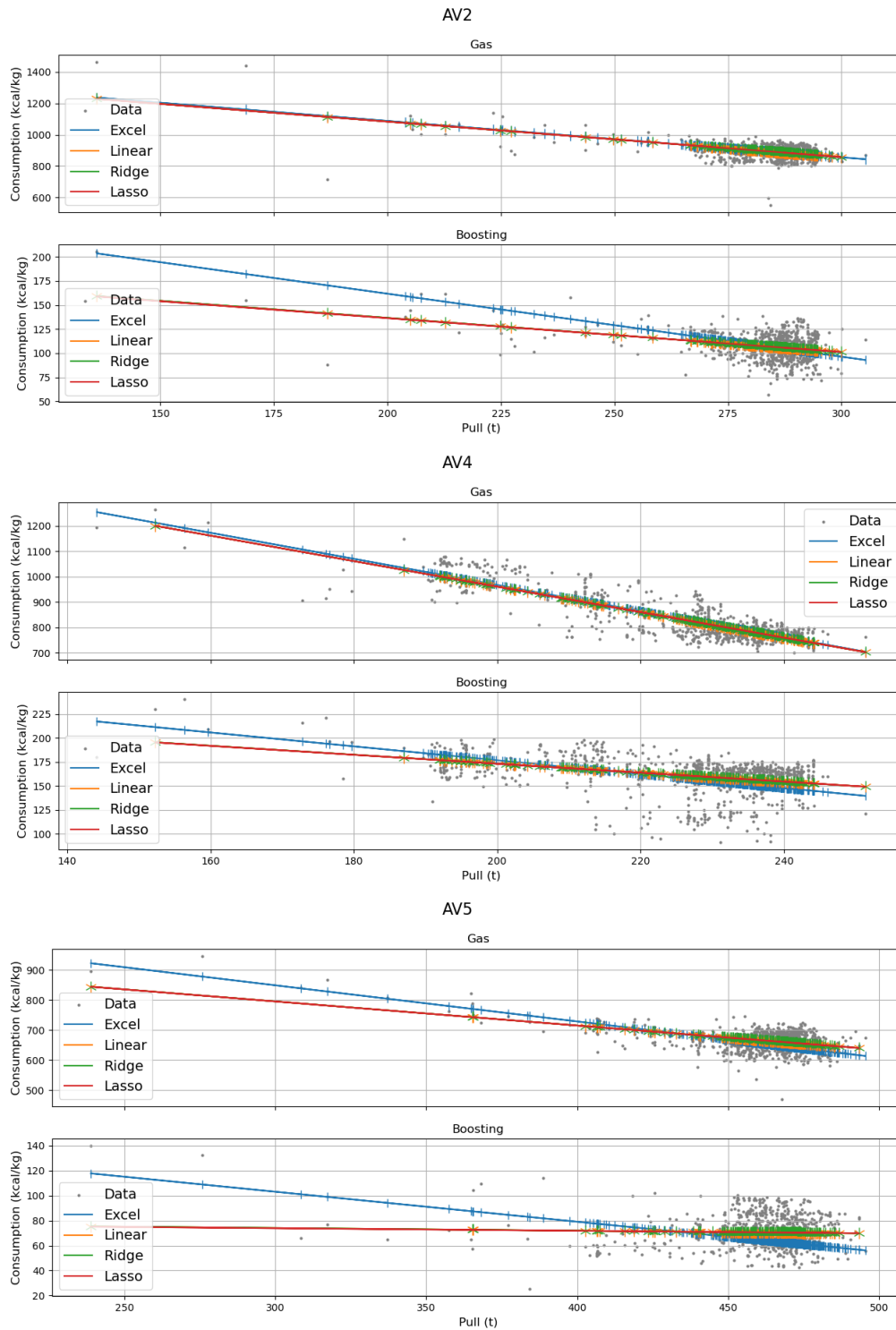
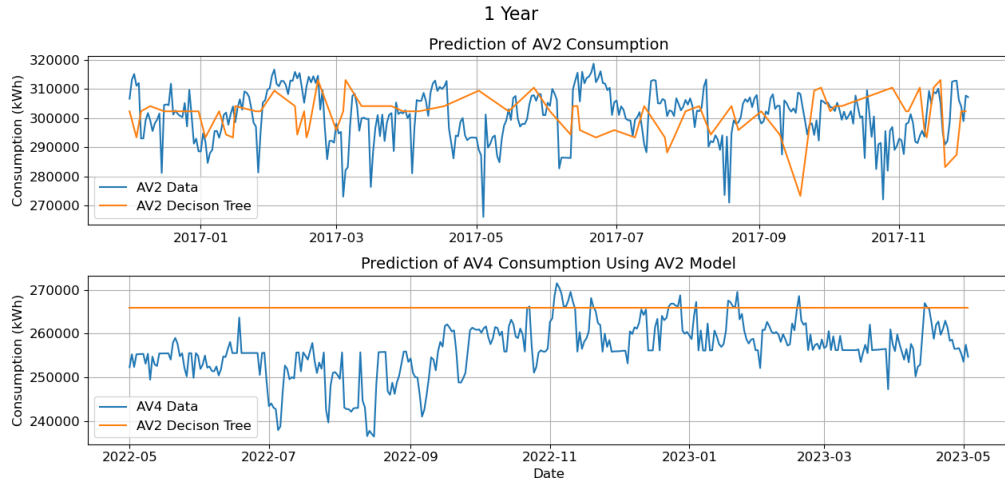
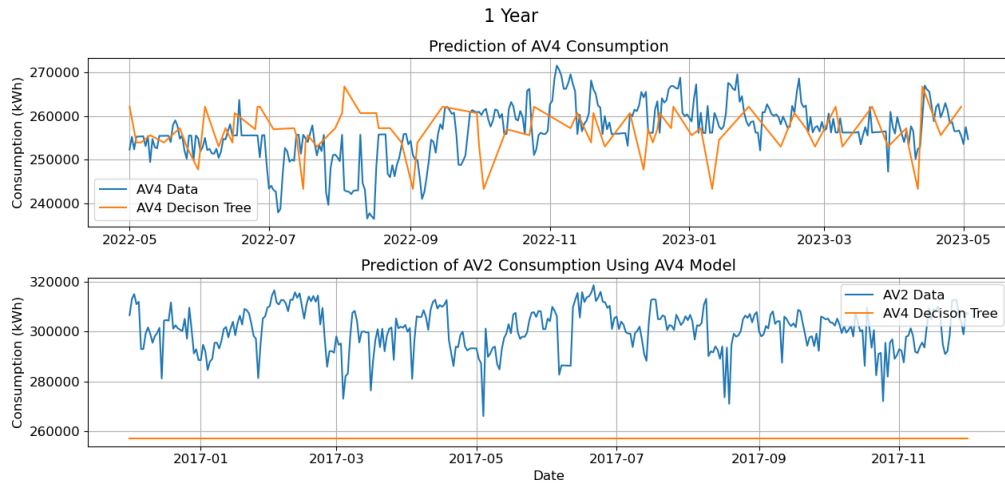


Figure A.7: Relationship between pull and specific consumption: regression performance comparison (4.3)

A.4 Transfer Learning: Comparison of Model Performance with Data Collected from a Different Furnace

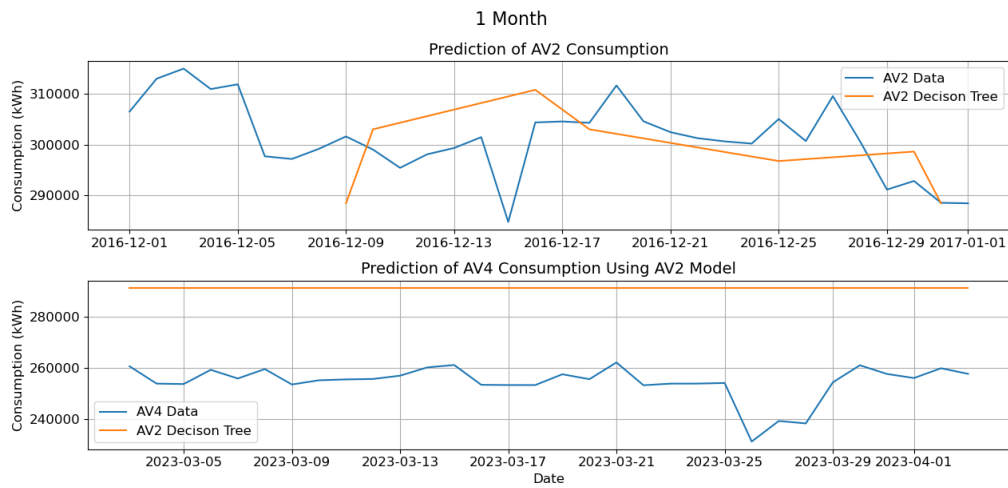


(a) Comparison of performance of AV2's model trained with AV2 and AV4 data

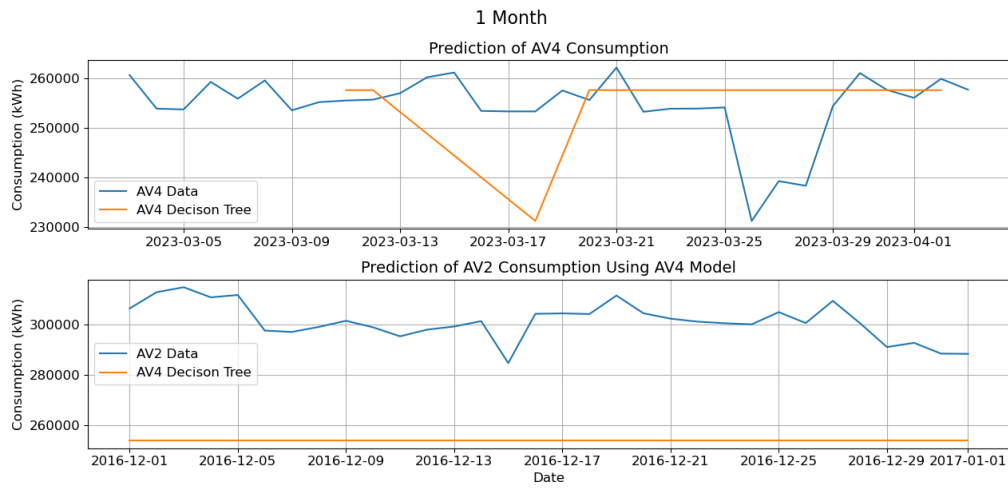


(b) Comparison of performance of AV4's model trained with AV2 and AV4 data

Figure A.8: Comparison of performance of one furnace's model with another furnace's data for a one-year interval (4.4)



(a) Comparison of performance of AV2’s model trained with AV2 and AV4 data



(b) Comparison of performance of AV4’s model trained with AV2 and AV4 data

Figure A.9: Comparison of performance of one furnace’s model with another furnace’s data for a one-month interval (4.4)

A.5 First Test Scenario Results

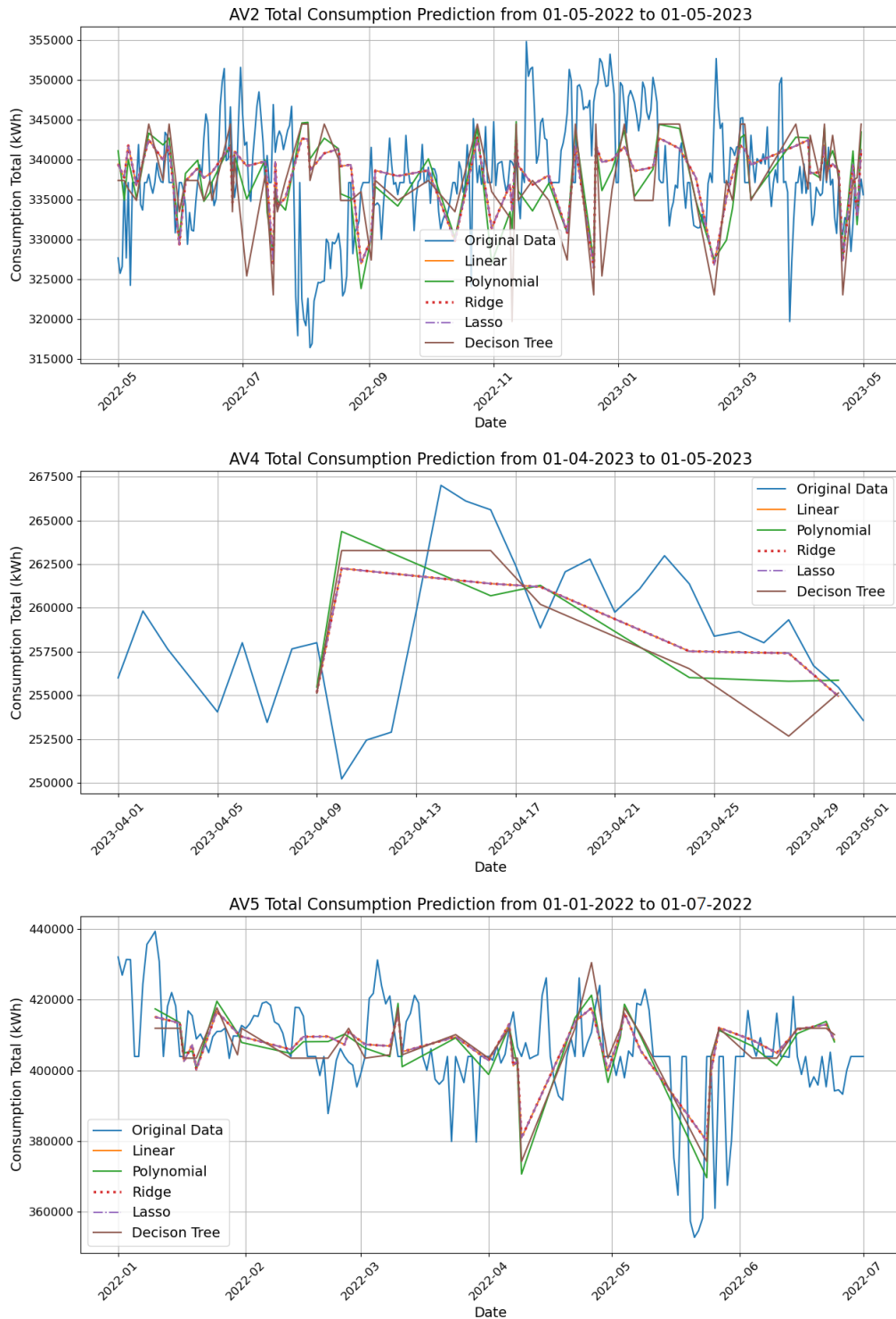


Figure A.10: Regression models performance (5.3)

Appendix B

Tables

Table B.1: Section of "Registros consumos_auto_1" Excel file (4.1)

	A	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	T	U	V	W	AA	AB	AC	AD
					AV2										AV4									
		Refiner AV2	F20	F21	F22	A20	A21	A22	Refiner AV4	F41	F42	F43	A41	A42	A43	Refiner AV5	F51	F52	F53	F54	A51	A52	A53	A54
																								(AV)
2		62125	25271	61204	26656	67955	101299	125184	790947	230840	237213	221272	33190	48402	76309	883850	819375	397628	449827	605281	137039	82350	167128	185354
2337	23/04/2023	62125	25271	61204	26656	67955	101299	125184	790947	230840	237213	221272	33190	48402	76309	883850	819375	397628	449827	605281	137039	82350	167128	185354
2338	24/04/2023	62211	25462	61445	26722	68226	101435	125414	794426	231425	237801	221884	33354	48523	76518	885677	820913	398330	450868	606169	137326	82396	167522	185905
2339	25/04/2023	62318	25662	61689	26789	68506	101584	125648	794426	232028	238397	222553	33524	48645	76734	887530	822459	399045	451922	607065	137616	82438	167944	186258
2340	26/04/2023	62415	25843	61933	26858	68775	101738	125883	796361	232636	238996	223135	33689	48767	76947	889376	824038	399769	452985	607979	137904	82476	168371	186484
2341	27/04/2023	62520	26054	62184	26927	69046	101878	126115	798159	233251	239605	223805	33854	48896	77234	891365	825627	400499	454053	608888	138184	82520	168995	186462
2342	28/04/2023	62635	26300	62432	26995	69314	102016	126340	799968	233842	240197	224450	34012	49025	77528	893129	827183	401137	455102	609790	138451	82645	169216	186736
2343	29/04/2023	62751	26547	62680	27064	69581	102156	126564	801777	234433	240789	225096	34169	49153	77823	894893	828738	401775	456151	610691	138717	82770	169638	187009
2344	30/04/2023	62866	26793	62928	27132	69849	102295	126788	803585	235024	241381	225741	34327	49281	78117	896656	830294	402412	457199	611593	138983	82895	170060	187283
2345	01/05/2023	62981	27040	63176	27201	70117	102434	127012	805394	235615	241973	226386	34485	49409	78411	898420	831849	403050	458248	612494	139250	83020	170482	187557
2346	02/05/2023	63098	27282	63418	27267	70578	102577	127233	807188	236193	242555	227021	34641	49532	78688	900031	833406	403715	459194	613466	139514	83188	170854	187827
2347	03/05/2023	63216	27538	63670	27346	70635	102736	127330	808980	236766	243138	227648	34798	49652	78972	901637	834957	404392	460094	614489	139783	83367	171185	188098
2348	04/05/2023	63322	27747	63916	27420	70894	102897	127550	810867	237315	243705	228265	34955	49776	79256	903311	836474	405079	460951	615459	140063	83556	171510	188372
2349	05/05/2023	63422	27973	64159	27493	71153	103058	127783	812704	237847	244261	228861	35108	49899	79527	904934	837972	405771	461789	616393	140327	83776	171812	188643
2350	06/05/2023	63529	28205	64400	27565	71412	103214	128012	814595	238396	244815	229469	35263	50022	79800	906434	839500	406459	462634	617368	140589	83996	172114	188912
2351	07/05/2023	63636	28439	64642	27656	71670	103370	128249	816503	238971	245368	230082	35417	50146	80066	907982	841012	407145	463467	618334	140849	84216	172420	189184
2352	08/05/2023	63747	28672	64885	27709	71925	103529	128491	818334	239517	245927	230690	35572	50273	80335	909571	842522	407842	464306	619289	141115	84437	172682	189454
2353	09/05/2023	63858	28904	65128	27782	72177	103689	128733	820191	240084	246482	231304	35729	50403	80596	911233	844036	408556	465143	620337	141376	84653	172946	189730
2354	10/05/2023	63974	29143	65370	27854	72430	103848	128973	822196	240665	247152	231920	35883	50471	80860	912775	845520	409243	465976	621388	141639	84870	173256	190002
2355	11/05/2023	64090	29383	65612	27925	72682	103999	129201	823992	241245	247670	232547	36034	50611	81123	914362	846989	409938	466779	622426	141884	85087	173568	190271
2356	12/05/2023	64204	29622	65852	27996	72935	104151	129420	825882	241802	248169	233164	36195	50740	81390	915936	848473	410636	467553	623479	142134	85291	173861	190537
2357	13/05/2023	64319	29865	66093	28068	73199	104305	129645	827732	242375	248666	233786	36357	50872	81655	917532	849955	411346	468326	624533	142400	85490	174142	190801
2358	14/05/2023	64432	30107	66334	28140	73444	104457	129860	829577	242942	249162	234407	36518	51002	81920	919071	851440	412056	469102	625594	142672	85691	174420	191066
2359	15/05/2023	64546	30349	66576	28212	73701	104612	130069	831391	243510	249662	235079	36679	51133	82215	920598	852954	412762	469876	626664	142916	85893	174710	191332
2360	16/05/2023	64658	30598	66818	28284	73959	104787	130280	833275	244097	250171	235649	36841	51285	82511	922155	854495	413474	470645	627751	143156	86091	174996	191596
2361	17/05/2023	64768	30852	67063	28355	74218	104951	130489	835214	244712	250692	236264	37002	51400	82791	923937	856026	414196	471579	628741	143397	86225	175304	191864
2362	18/05/2023	64879	31099	67309	28428	74478	105136	130700	837191	245364	251292	236893	37170	51559	83092	925744	857610	414917	472591	629741	143647	86321	175624	192138
2363	19/05/2023	64985	31335	67551	28498	74730	105293	130913	839156	245996	251888	237505	37333	51671	83379	927219	859209	415649	473478	630723	143896	86521	175957	192406

Table B.3: Metrics obtained for various regression models for each furnace (5.3)

		1 MONTH From '2022-01-01' to '2022-02-01'					1 MONTH From '2023-04-01' to '2023-05-01'				
		R ²	MAE	Train RMSE	Test RMSE	SD	R ²	MAE	Train RMSE	Test RMSE	SD
AV2	Linear	-0.51722	1962.36	4068.27	2218.44	1801.04	-0.0579	2063.04	2480.62	2573.18	2501.78
	Polynomial	-0.77857	1886.64	3503.34	2401.93		-0.5109	2680.91	2044.47	3075.15	
	Ridge	-0.51722	1962.36	4068.27	2218.44		-0.0579	2063.04	2480.62	2573.18	
	Lasso	-0.51740	1962.50	4068.27	2218.58		-0.0578	2062.88	2480.62	2573.12	
	Decision Tree*	-5.27499	3852.25	1755.27	4511.60		-0.0213	2051.77	740.56	2528.26	
AV4	Linear	-0.12991	3019.26	2355.41	4627.11	4353.00	0.3733	2581.70	2488.19	3511.07	4435.18
	Polynomial	0.09618	3020.39	2270.73	4138.37		0.5781	2312.25	1988.63	2880.77	
	Ridge	-0.12991	3019.26	2355.41	4627.11		0.3733	2581.69	2488.19	3511.07	
	Lasso	-0.13007	3019.28	2355.41	4627.44		0.3733	2581.60	2488.19	3511.04	
	Decision Tree*	-0.37746	3313.12	944.53	5108.91		0.7329	2006.36	1142.34	2292.02	
AV5	Linear	0.54935	5895.09	4288.90	7725.89	11508.80	0.0658	10036.15	11601.52	11079.42	11462.79
	Polynomial	0.78358	4150.30	3164.43	5354.03		-1.3121	14688.28	8699.73	17429.85	
	Ridge	0.54935	5895.09	4288.90	7725.89		0.0658	10036.15	11601.52	11079.42	
	Lasso	0.54935	5895.09	4288.90	7725.89		0.0658	10036.11	11601.52	11079.35	
	Decision Tree*	0.69533	4295.48	1490.26	6352.55		-0.6617	10931.75	2427.71	14776.31	
		6 MONTHS From '2022-01-01' to '2022-07-01'					6 MONTHS From '2022-11-01' to '2023-05-01'				
		R ²	MAE	Train RMSE	Test RMSE	SD	R ²	MAE	Train RMSE	Test RMSE	SD
AV2	Linear	0.12652	3667.54	5791.65	4879.91	5221.39	0.2725	4427.67	4848.21	5396.29	6326.92
	Polynomial	0.13571	3944.17	5410.93	4854.17		0.3389	3867.19	4542.02	5144.42	
	Ridge	0.12652	3667.54	5791.65	4879.91		0.2725	4427.67	4848.21	5396.29	
	Lasso	0.12652	3667.54	5791.65	4879.91		0.2726	4427.65	4848.21	5396.18	
	Decision Tree*	0.14104	2898.33	4382.21	4839.19		0.7045	2704.00	3106.45	3439.46	
AV4	Linear	0.41027	3780.03	4566.89	4774.53	6217.36	0.6687	2773.69	3631.51	3439.31	5975.30
	Polynomial	0.50911	3567.16	4073.18	4356.09		0.6822	2565.18	3314.17	3368.44	
	Ridge	0.41027	3780.03	4566.89	4774.53		0.6687	2773.69	3631.51	3439.31	
	Lasso	0.41027	3780.04	4566.89	4774.57		0.6687	2773.69	3631.51	3439.31	
	Decision Tree*	0.35276	3843.70	2857.58	5001.95		0.1495	3047.94	2312.95	5510.54	
AV5	Linear	-0.18796	7187.84	9912.44	9352.19	8580.51	0.0766	6488.44	8144.87	8217.12	8550.95
	Polynomial	-0.67325	8204.48	8491.60	11099.25		0.1830	6093.71	7946.48	7729.23	
	Ridge	-0.18796	7187.84	9912.44	9352.19		0.0766	6488.44	8144.87	8217.12	
	Lasso	-0.18795	7187.83	9912.44	9352.17		0.0766	6488.44	8144.87	8217.12	
	Decision Tree*	-0.42003	7722.83	6334.22	10224.96		-0.2433	5565.95	5323.83	9534.70	
		1 YEAR From '2022-01-01' to '2023-01-01'					1 YEAR From '2022-05-01' to '2023-05-01'				
		R ²	MAE	Train RMSE	Test RMSE	SD	R ²	MAE	Train RMSE	Test RMSE	SD
AV2	Linear	0.27653	4617.79	5306.86	5634.38	6624.25	0.4410	3894.18	5363.03	4880.07	6527.16
	Polynomial	0.34722	4247.86	5172.26	5352.03		0.4296	3980.94	4982.61	4929.72	
	Ridge	0.27653	4617.79	5306.86	5634.38		0.4410	3894.18	5363.03	4880.07	
	Lasso	0.27654	4617.78	5306.86	5634.36		0.4410	3894.17	5363.03	4880.07	
	Decision Tree*	0.53392	3417.33	4480.53	4522.38		0.4935	3535.43	4202.58	4645.49	
AV4	Linear	0.49801	4407.92	5224.59	5362.46	7568.63	0.4147	3774.71	4415.12	4487.32	5865.36
	Polynomial	0.61827	3531.16	4156.50	4676.21		0.4352	3715.92	4319.25	4407.89	
	Ridge	0.49801	4407.92	5224.59	5362.46		0.4147	3774.71	4415.12	4487.32	
	Lasso	0.49801	4407.95	5224.59	5362.49		0.4147	3774.71	4415.12	4487.32	
	Decision Tree*	0.65405	3187.04	3393.95	4451.69		0.4224	3364.53	2952.07	4457.67	
AV5	Linear	0.00672	9327.25	9669.92	12116.79	12157.69	0.3761	7667.59	9594.65	9437.93	11949.01
	Polynomial	0.26734	8227.85	8725.11	10406.44		0.3014	8277.92	9250.15	9987.57	
	Ridge	0.00672	9327.25	9669.92	12116.79		0.3761	7667.59	9594.65	9437.93	
	Lasso	0.00672	9327.23	9669.92	12116.77		0.3761	7667.58	9594.65	9437.93	
	Decision Tree*	0.44357	7475.21	7307.51	9068.93		0.3382	7653.17	7093.32	9720.49	

*max_depth = 4

Table B.4: AV2's results obtained from the second test scenario (5.3)

Model: Polynomial		180												365																																																																																																											
		7				15				30				60				7				15				30				60																																																																																											
Prediction Duration	Prediction Period	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90																																																																														
r ²		-0.1385	-0.0686	-0.0602	-0.1860	-0.0748	-0.0294	-1.1460	-0.0921	-0.0857	-0.3558	-0.3053	-0.2237	-0.6521	-0.1023	-0.0959	-0.7809	-0.2414	-0.2337	-5.9719	-0.2026	-0.3574	-2.9628	-0.5423	-0.6525	48501	43518	37756	49237	43279	37737	63027	43835	38257	51549	45887	40221	34214	29035	27108	36866	30179	28334	49908	30103	29026	52364	35742	33343	58988	53378	48131	58711	53627	48555	55448	53024	48053	56542	50435	48047	34689	34302	32778	34857	34497	32867	34751	34637	33156	33835	34368	33039	67653	58548	52044	68802	68518	51119	93594	59642	53073	73105	64079	55371	48963	37359	34904	50836	39646	37034	101200	39261	39084	80927	47174	45767	63404	56638	50545	63178	56445	50383	63891	57073	50935	62785	56086	50055	38093	35583	33342	38093	35583	33342	38327	35801	33546	40653	37985	35603
MAE		48501	43518	37756	49237	43279	37737	63027	43835	38257	51549	45887	40221	34214	29035	27108	36866	30179	28334	49908	30103	29026	52364	35742	33343	58988	53378	48131	58711	53627	48555	55448	53024	48053	56542	50435	48047	34689	34302	32778	34857	34497	32867	34751	34637	33156	33835	34368	33039	67653	58548	52044	68802	68518	51119	93594	59642	53073	73105	64079	55371	48963	37359	34904	50836	39646	37034	101200	39261	39084	80927	47174	45767	63404	56638	50545	63178	56445	50383	63891	57073	50935	62785	56086	50055	38093	35583	33342	38093	35583	33342	38327	35801	33546	40653	37985	35603																								
Training RMSE		58988	53378	48131	58711	53627	48555	55448	53024	48053	56542	50435	48047	34689	34302	32778	34857	34497	32867	34751	34637	33156	33835	34368	33039	67653	58548	52044	68802	68518	51119	93594	59642	53073	73105	64079	55371	48963	37359	34904	50836	39646	37034	101200	39261	39084	80927	47174	45767	63404	56638	50545	63178	56445	50383	63891	57073	50935	62785	56086	50055	38093	35583	33342	38093	35583	33342	38327	35801	33546	40653	37985	35603																																																
Testing RMSE		67653	58548	52044	68802	68518	51119	93594	59642	53073	73105	64079	55371	48963	37359	34904	50836	39646	37034	101200	39261	39084	80927	47174	45767	63404	56638	50545	63178	56445	50383	63891	57073	50935	62785	56086	50055	38093	35583	33342	38093	35583	33342	38327	35801	33546	40653	37985	35603																																																																								
SD		63404	56638	50545	63178	56445	50383	63891	57073	50935	62785	56086	50055	38093	35583	33342	38093	35583	33342	38327	35801	33546	40653	37985	35603	63404	56638	50545	63178	56445	50383	63891	57073	50935	62785	56086	50055	38093	35583	33342	38093	35583	33342	38327	35801	33546	40653	37985	35603																																																																								

Model: Decision Tree (max_depth = 4)		180												365																																																																																																											
		7				15				30				60				7				15				30				60																																																																																											
Prediction Duration	Prediction Period	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90																																																																														
r ²		-0.0230	0.1034	0.0626	-0.0875	-0.0085	0.0912	-0.2282	-0.1710	-0.1386	-0.3139	-0.0054	-0.1623	-0.2229	-0.1244	-0.0819	-0.4896	-0.2059	-0.1937	-0.5533	-0.2686	-0.2657	-0.0666	-0.0444	-0.0326	66294	59475	51232	66004	60766	51064	72621	69255	57829	78149	60425	58165	46868	41450	38580	53329	44231	41172	53463	44326	42577	75294	64570	60225	57086	57086	54377	55799	56843	54925	50421	56206	53452	52137	56689	51668	32374	34804	36025	32142	34600	36218	31048	34215	36337	30516	34224	34967	102353	85786	78793	105102	90622	77284	113002	98787	87501	114855	89953	86878	67647	60652	55814	74660	62811	58628	76714	64822	60744	251806	248142	245898	101196	90600	81382	100783	90239	81067	101966	91290	82003	100200	89709	80583	61172	57197	53660	61172	57197	53660	61552	57552	53992	243814	242813	241982
MAE		66294	59475	51232	66004	60766	51064	72621	69255	57829	78149	60425	58165	46868	41450	38580	53329	44231	41172	53463	44326	42577	75294	64570	60225	57086	57086	54377	55799	56843	54925	50421	56206	53452	52137	56689	51668	32374	34804	36025	32142	34600	36218	31048	34215	36337	30516	34224	34967	102353	85786	78793	105102	90622	77284	113002	98787	87501	114855	89953	86878	67647	60652	55814	74660	62811	58628	76714	64822	60744	251806	248142	245898	101196	90600	81382	100783	90239	81067	101966	91290	82003	100200	89709	80583	61172	57197	53660	61172	57197	53660	61552	57552	53992	243814	242813	241982																								
Training RMSE		57086	57086	54377	55799	56843	54925	50421	56206	53452	52137	56689	51668	32374	34804	36025	32142	34600	36218	31048	34215	36337	30516	34224	34967	102353	85786	78793	105102	90622	77284	113002	98787	87501	114855	89953	86878	67647	60652	55814	74660	62811	58628	76714	64822	60744	251806	248142	245898	101196	90600	81382	100783	90239	81067	101966	91290	82003	100200	89709	80583	61172	57197	53660	61172	57197	53660	61552	57552	53992	243814	242813	241982																																																
Testing RMSE		102353	85786	78793	105102	90622	77284	113002	98787	87501	114855	89953	86878	67647	60652	55814	74660	62811	58628	76714	64822	60744	251806	248142	245898	101196	90600	81382	100783	90239	81067	101966	91290	82003	100200	89709	80583	61172	57197	53660	61172	57197	53660	61552	57552	53992	243814	242813	241982																																																																								
SD		101196	90600	81382	100783	90239	81067	101966	91290	82003	100200	89709	80583	61172	57197	53660	61172	57197	53660	61552	57552	53992	243814	242813	241982	101196	90600	81382	100783	90239	81067	101966	91290	82003	100200	89709	80583	61172	57197	53660	61172	57197	53660	61552	57552	53992	243814	242813	241982																																																																								

Table B.5: AV4's results obtained from the second test scenario (5.3)

Model: Polynomial		180												365											
Prediction Duration	Prediction Period	7						15						30						60					
Training Period		30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90
r^2		-0.2742	0.5253	0.4881	-0.3793	0.4613	0.4833	-0.1321	0.2134	0.3153	0.1551	0.3207	0.2663	0.2498	0.4469	0.4889	0.1112	0.3695	0.3466	0.1809	0.4141	0.4551	0.2462	0.2139	0.4113
MAE		3493	2978	3222	3769	3100	3221	4125	3810	3838	3759	3572	3916	3658	3745	3598	4120	3783	3837	4496	3892	3941	4350	4536	4017
Training RMSE		3055	3385	3816	3008	3379	2938	2938	3385	3879	2193	3656	4097	3468	3917	4028	3544	3898	4100	3568	3785	3785	4032	3294	3994
Testing RMSE		6354	3895	4129	6528	4097	4097	5973	5000	4760	5179	4665	4950	5814	5008	4821	6328	5347	5451	6042	5127	4951	5749	5885	5098
SD		5629	5654	5771	5559	5583	5699	5614	5637	5752	5635	5660	5779	6712	6734	6743	6712	6734	6743	6676	6698	6707	6621	6637	6645

Model: Decision Tree (max_depth = 4)		180												365											
Prediction Duration	Prediction Period	7						15						30						60					
Training Period		30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90	30	60	90
r^2		0.4244	0.4030	0.4434	0.4007	0.3795	0.4471	0.1179	0.2503	0.2049	0.1489	0.0787	0.1104	0.4262	0.4924	0.5073	0.5022	0.5485	0.5450	0.3850	0.4431	0.4550	0.0055	0.1845	0.3629
MAE		2898	3079	3143	3043	3191	3200	3535	3508	3917	3536	4021	4023	3636	3488	3471	3474	3315	3444	3842	3712	3756	4812	4567	4149
Training RMSE		2033	2240	2676	2082	2221	2676	1955	2450	2773	1659	2475	2881	2098	2598	3057	2310	2595	2880	2254	2616	3004	2414	2674	2916
Testing RMSE		4283	4380	4318	4316	4410	4250	5287	4894	5144	5214	5448	5466	5084	4797	4733	4738	4528	4551	5239	5001	4954	6623	6012	5319
SD		5646	5669	5788	5575	5598	5716	5630	5653	5769	5652	5676	5795	6712	6734	6743	6716	6738	6747	6680	6702	6711	6642	6657	6664

Table B.6: AV5's results obtained from the second test scenario (5.3)

Model: Polynomial												
Prediction Duration Period	180						365					
	7		15		60		7		15		60	
Training Period	30	60	90	30	60	90	30	60	90	30	60	90
r^2	-0.1779	-0.2136	-0.2068	-0.6049	-0.9363	-0.7538	-0.8450	-2.1686	-1.1706	-0.6873	-0.2715	-1.0971
MAE	42601	37810	33654	46368	42340	36652	48405	50287	38824	49932	38973	37579
Training RMSE	41929	38099	34756	42301	38351	35125	41945	37941	35007	42566	38262	35780
Testing RMSE	49611	44781	39165	57654	56334	47039	62264	72565	52678	59176	45685	51465
SD	45711	40650	35652	45510	40483	35520	45839	40766	35756	45557	40516	35539

Model: Decision Tree (max_depth = 4)												
Prediction Duration Period	180						365					
	7		15		60		7		15		60	
Training Period	30	60	90	30	60	90	30	60	90	30	60	90
r^2	-0.3005	-0.1218	-0.2001	-0.3552	-0.1062	-0.1893	-0.3375	-0.3890	-0.2383	-0.5479	-0.3960	-0.2651
MAE	76646	67051	64045	78950	66467	64468	78770	76936	65628	86951	79859	65768
Training RMSE	52573	53281	51746	53652	53523	51948	52731	51369	51362	53493	51233	50699
Testing RMSE	100243	83051	76430	101746	82016	75683	101912	92645	77841	108966	92312	78195
SD	87901	78411	69769	87402	77979	69398	88121	78610	69950	87582	78128	69520

Model: Lasso												
Prediction Duration Period	180						365					
	7		15		60		7		15		60	
Training Period	30	60	90	30	60	90	30	60	90	30	60	90
r^2	-0.1390	-0.0857	-0.0634	-0.3718	-0.1706	-0.0576	-0.4621	-0.0990	-0.0548	-0.7272	-0.3662	-0.1222
MAE	41763	37500	32905	43514	38024	32774	45697	38454	32632	49433	40189	31868
Training RMSE	42426	38500	34842	42820	38795	35165	42225	38719	35104	42562	39161	35735
Testing RMSE	48785	42356	36765	53303	43801	36528	55428	42736	36723	59872	47357	37647
SD	45711	40650	35652	45510	40483	35520	45839	40766	35756	45557	40516	35539

Table B.7: Results from the final application (6.1)

Date	Planned Pull (kg)	Gas (kcal/kg)	Boosting (kcal/kg)	Total (kcal/kg)	Gas (Nm ³)	Gas (€)	Boosting (kWh)	Electricity (€)
2023-06-18	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-19	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-20	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-21	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-22	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-23	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-24	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-25	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-26	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-27	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-28	231317.28	769.963643	161.556240	931.519883	19597.233734	222.531219	43454.360419	4345.436042
2023-06-29	229560.48	779.878936	165.111962	944.916884	19696.976796	223.663825	44168.466578	4407.346658
2023-06-30	229560.48	779.804922	165.111962	944.916884	19696.976796	223.663825	44168.466578	4407.346658
2023-07-01	229560.48	779.804922	165.111962	944.916884	19696.976796	223.663825	44168.466578	4407.346658
2023-07-02	229560.48	779.804922	165.111962	944.916884	19696.976796	223.663825	44073.466578	4407.346658

Appendix C

Code

C.1 C# Script for Detection of Flow Meter Malfunctions (3.2)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using inray.OPCRouter.ScriptPlugIn.Shared;
5 using inray.OPCRouter.ScriptPlugIn.Runtime;
6 using inray.OPCRouter.ScriptPlugIn.Runtime.TransferObject;
7
8 namespace OPCRouter.Script
9 {
10     public class Threshold_20 : ScriptTransferObjectBase
11     {
12         public double previousValue;
13         private const double ThresholdPercentage = 0.2;
14
15         /// Method is being called after another transfer object
16         /// wrote values to this transfer object
17         public override void Write()
18         {
19             // Calculate the threshold value
20             double thresholdValue = previousValue * ThresholdPercentage;
21             // Compare the absolute difference to the threshold value
22             if (Math.Abs(previousValue - currentValue) > thresholdValue)
23             { // If the difference is over the threshold
24                 previousValue = currentValue;
25                 alarm = true;
26             }
27             else { // If the difference is below or equal to the threshold
28                 alarm = false;
29             }
30         }
31     }
32 }
```

C.2 Final Application (6.1)

```

1 import datetime
2 import pandas as pd
3 import matplotlib.pyplot as plt
4
5 from sklearn.tree import DecisionTreeRegressor
6 from sklearn.model_selection import GridSearchCV
7 from sklearn.ensemble import BaggingRegressor
8
9 GAS_kWhEur = 0.12
10 ELE_kWhEur = 0.10
11
12 cutoff_date = pd.to_datetime(datetime.date.today() - datetime.timedelta(days=180))
13
14 # Save registros_auto data to a dataframe
15 reg_data = pd.read_excel(r"L:\Energia\Registros consumos_auto_1.xlsx",
16                         skiprows=1, sheet_name='INPUT', parse_dates=['Dia'])
17 reg_data = reg_data.loc[reg_data['Dia'] >= cutoff_date]
18 reg_data = reg_data.loc[:, ['Dia', 'Forno 4', 'Kwh.1', 'Nm3.1', 'kWh/m3']]
19 reg_data = reg_data.rename(columns=lambda x: x.strip())
20 reg_data = reg_data.rename(
21     columns={
22         'Dia': 'Date',
23         'Forno 4': 'Pull',
24         'Kwh.1': 'Boosting',
25         'Nm3.1': 'Gas',
26         'kWh/m3': 'REN'
27     }
28 )
29 reg_data = reg_data.dropna()
30
31 reg_data['Gas_kcal/kg'] = reg_data['Gas'] * reg_data['REN'] * 860 / reg_data['Pull']
32 reg_data['Boosting_kcal/kg'] = reg_data['Boosting'] * 860 / reg_data['Pull']
33 reg_data['Total_kcal/kg'] = reg_data['Boosting_kcal/kg'] + reg_data['Gas_kcal/kg']
34
35 # Filter data
36 for i in range(1, len(reg_data)):
37     current_value = reg_data.iloc[i, reg_data.columns.get_loc('Pull')]
38     previous_value = reg_data.iloc[i - 1, reg_data.columns.get_loc('Pull')]
39     if abs(current_value - previous_value) > 0.08 * previous_value:
40         reg_data.iloc[i, reg_data.columns.get_loc('Pull')] = reg_data['Pull'].mean()
41         reg_data.iloc[i, reg_data.columns.get_loc('Gas_kcal/kg')] = reg_data['
42 Gas_kcal/kg'].mean()
43         reg_data.iloc[i, reg_data.columns.get_loc('Boosting_kcal/kg')] = reg_data['
44 Boosting_kcal/kg'].mean()
45     current_value = reg_data.iloc[i, reg_data.columns.get_loc('Total_kcal/kg')]
46     previous_value = reg_data.iloc[i - 1, reg_data.columns.get_loc('Total_kcal/kg')]
47     if abs(current_value - previous_value) > 0.023 * previous_value:

```

```

46     reg_data.iloc[i,reg_data.columns.get_loc('Total_kcal/kg')] = reg_data['
Total_kcal/kg'].mean()
47     reg_data.iloc[i,reg_data.columns.get_loc('Gas_kcal/kg')] = reg_data['
Gas_kcal/kg'].mean()
48     reg_data.iloc[i,reg_data.columns.get_loc('Boosting_kcal/kg')] = reg_data['
Boosting_kcal/kg'].mean()
49
50 # Save planned pull data to a dataframe
51 pull_data = pd.read_excel(r"C:\Users\rmonte\Desktop\Tiragens.xlsx", skiprows=1,
sheet_name='AV4')
52 pull_data = pull_data.loc[:,['Date','Tir.3']]
53 pull_data = pull_data.rename(columns={'Tir.3': 'Pull'})
54 pull_data = pull_data.dropna()
55 pull_data['Pull'] = pull_data['Pull']*1000
56
57 days_to_subtract = (datetime.date.today().weekday()+2) % 7 # get date from sunday
to sunday
58
59 # Define the specified date
60 date = datetime.date.today() - datetime.timedelta(days=days_to_subtract)
61
62 # Define the start and end dates
63 pred_start_date = date + datetime.timedelta(days=1)
64 pred_end_date = pred_start_date + datetime.timedelta(days=14)
65
66 # Prepare an empty DataFrame to store the predictions
67 pred_gas = pd.DataFrame(columns=['Date', 'Prediction'])
68 pred_ele = pd.DataFrame(columns=['Date', 'Prediction'])
69
70 # Convert prediction dates to datetime objects
71 pred_start_date = pd.to_datetime(pred_start_date)
72 pred_end_date = pd.to_datetime(pred_end_date)
73 date_range = pd.date_range(start=pred_start_date, end=pred_end_date)
74
75 # Prepare the prediction data for the current week
76 planned_pull = pull_data[pull_data['Date'].isin(date_range)]
77
78 # Filter the training data for the current prediction week
79 training_data = reg_data[(reg_data['Date'] >= pred_start_date - pd.DateOffset(days
=90)) & (reg_data['Date'] < pred_start_date)]
80
81 # Split the data into input and output variables
82 X_train = training_data[['Pull']]
83 y_train_gas = training_data[['Gas_kcal/kg']]
84 y_train_ele = training_data[['Boosting_kcal/kg']]
85
86 # Create the Decision Tree Regressors
87 base_model_gas = DecisionTreeRegressor(max_depth=4)
88 base_model_ele = DecisionTreeRegressor(max_depth=4)

```

```

89 # Define the BaggingRegressor
90 bag_gas = BaggingRegressor(estimator=base_model_gas, random_state=42)
91 bag_ele = BaggingRegressor(estimator=base_model_ele, random_state=42)
92 # Define the hyperparameters to search over
93 param_grid = {'n_estimators': [10, 20, 30]}
94 # Create the grid search object
95 grid_gas = GridSearchCV(estimator=bag_gas, param_grid=param_grid, cv=5, scoring='
    neg_mean_squared_error')
96 grid_ele = GridSearchCV(estimator=bag_ele, param_grid=param_grid, cv=5, scoring='
    neg_mean_squared_error')
97 # Fit it to the data
98 grid_gas.fit(X_train, y_train_gas.values.ravel())
99 grid_ele.fit(X_train, y_train_ele.values.ravel())
100 # Get the best estimator
101 model_gas = grid_gas.best_estimator_
102 model_ele = grid_ele.best_estimator_
103 # Train the model with the best estimator
104 model_gas.fit(X_train, y_train_gas.values.ravel())
105 model_ele.fit(X_train, y_train_ele.values.ravel())
106
107 pred_gas = model_gas.predict(planned_pull[['Pull']])
108 pred_ele = model_ele.predict(planned_pull[['Pull']])
109
110 # Store the prediction in the predictions DataFrame
111 pred_gas = pd.DataFrame({'Date': pull_data[pull_data['Date'].isin(date_range)][
    'Date'], 'Prediction': pred_gas.flatten()})
112 pred_ele = pd.DataFrame({'Date': pull_data[pull_data['Date'].isin(date_range)][
    'Date'], 'Prediction': pred_ele.flatten()})
113
114 results = pd.DataFrame(columns=['Date', 'Planned Pull (kg)', 'Gas (kcal/kg)', '
    Boosting (kcal/kg)'])
115
116 results = pd.DataFrame({'Date': pull_data[pull_data['Date'].isin(date_range)][
    'Date'], 'Planned Pull (kg)': pull_data[pull_data['Date'].isin(date_range)][
    'Pull'], 'Gas (kcal/kg)': pred_gas['Prediction'], 'Boosting (kcal/kg)': pred_ele[
    'Prediction']})
117
118 results['Total (kcal/kg)']=results['Gas (kcal/kg)'] +results['Boosting (kcal/kg)']
119 results['Gas (Nm^3)']=results['Gas (kcal/kg)']*results['Planned Pull (kg)']/
    (reg_data['REN'].mean()*860)
120 results['Gas expense (Eur)']=results['Gas (Nm^3)']/reg_data['REN'].mean()*
    GAS_kWhEur
121 results['Boosting (kWh)']=results['Boosting (kcal/kg)'] *results['Planned Pull (kg)
    ']/860
122 results['Electricity expense (Eur)']=results['Boosting (kWh)']*ELE_kWhEur
123
124 # Calculate standard deviations
125 a = reg_data['Gas_kcal/kg'].std()/reg_data['Gas_kcal/kg'].mean()
126 b = reg_data['Boosting_kcal/kg'].std()/reg_data['Boosting_kcal/kg'].mean()

```

```

127 c = reg_data['Total_kcal/kg'].std()/reg_data['Total_kcal/kg'].mean()
128
129 display_range = pd.date_range(start=datetime.date.today()-pd.DateOffset(days=30),
    end=datetime.date.today())
130
131 # Display the predictions
132 print(results)
133
134 fig, (ax1, ax2, ax3) = plt.subplots(3, 1, sharex=True, figsize=(10, 14))
135 ax1.plot(reg_data[reg_data['Date'].isin(display_range)]['Date'],
136         reg_data[reg_data['Date'].isin(display_range)]['Gas_kcal/kg'], label='
    Previous Data')
137 ax1.plot(results['Date'], results['Gas (kcal/kg)'], label='Prediction')
138 ax1.fill_between(results['Date'], (1+a)*results['Gas (kcal/kg)'],
139                (1-a)*results['Gas (kcal/kg)'], color='orange', alpha=0.3)
140 ax1.set_ylabel('Gas Consumption (kcal/kg)', fontsize=14)
141 ax1.tick_params(axis='both', which='major', labelsize=12)
142 ax1.legend(fontsize=14)
143 ax1.grid()
144
145 ax2.plot(reg_data[reg_data['Date'].isin(display_range)]['Date'],
146         reg_data[reg_data['Date'].isin(display_range)]['Boosting_kcal/kg'], label='
    Previous Data')
147 ax2.plot(results['Date'], results['Boosting (kcal/kg)'], label='Prediction')
148 ax2.fill_between(results['Date'], (1+b)*results['Boosting (kcal/kg)'],
149                (1-b)*results['Boosting (kcal/kg)'], color='orange', alpha=0.3)
150 ax2.set_ylabel('Boosting Consumption (kcal/kg)', fontsize=14)
151 ax2.tick_params(axis='both', which='major', labelsize=12)
152 ax2.legend(fontsize=14)
153 ax2.grid()
154
155 ax3.plot(reg_data[reg_data['Date'].isin(display_range)]['Date'],
156         reg_data[reg_data['Date'].isin(display_range)]['Total_kcal/kg'], label='
    Previous Data')
157 ax3.plot(results['Date'], results['Total (kcal/kg)'], label='Prediction')
158 ax3.fill_between(results['Date'], (1+c)*results['Total (kcal/kg)'],
159                (1-c)*results['Total (kcal/kg)'], color='orange', alpha=0.3)
160 ax3.set_xlabel('Date', fontsize=14)
161 ax3.set_ylabel('Total Consumption (kcal/kg)', fontsize=14)
162 ax3.tick_params(axis='both', which='major', labelsize=12)
163 ax3.legend(fontsize=14)
164 ax3.grid()
165 plt.tight_layout()
166 plt.show()

```


References

- [1] Antonio Calderón Godoy and Isaías González Pérez. Integration of Sensor and Actuator Networks and the SCADA System to Promote the Migration of the Legacy Flexible Manufacturing System towards the Industry 4.0 Concept. *Journal of Sensor and Actuator Networks*, 7(2):23, May 2018. URL: <http://www.mdpi.com/2224-2708/7/2/23>, doi:10.3390/jsan7020023.
- [2] Marcel Nicola, Claudiu-Ionel Nicola, Marian Duță, and Sacerdo Dumitru. SCADA Systems Architecture Based on OPC and Web Servers and Integration of Applications for Industrial Process Control. *International Journal of Control Science and Engineering*, 8(1):13–21, 2018. doi:10.5923/j.control.20180801.02.
- [3] Jorge Fernandes Alves. *BA · Marca com história no vidro de embalagem*. BA Vidro, S.A., 2012.
- [4] João Pedro Alves. Raw Materials Process Management Improvement in the Glass Industry. Master's thesis, University of Porto, January 2023.
- [5] European Commission. Industry 5.0, 2023. Last visited on 29-06-2023. URL: https://research-and-innovation.ec.europa.eu/research-area/industrial-research-and-innovation/industry-50_en.
- [6] Michiyuki Yagi and Shunsuke Managi. The spillover effects of rising energy prices following 2022 Russian invasion of Ukraine. *Economic Analysis and Policy*, 77:680–695, March 2023. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0313592622002338>, doi:10.1016/j.eap.2022.12.025.
- [7] EUROSTAT. Statistics explained, April 2023. Last visited on 12-06-2023. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Main_Page.
- [8] European Committee for Standardization. ISO 9001 Sistemas de Gestão da Qualidade, 2015.
- [9] ISO 22000 Food Safety Management Systems, 2018.
- [10] European Committee for Standardization. ISO 14001 Sistemas de Gestão Ambiental, 2015.
- [11] Social Accountability International. ISO 8000 Social Accountability, 2014.
- [12] Project Committee ISO/PC 283. ISO 45001 Sistemas de Gestão da Segurança e Saúde no Trabalho, 2019.
- [13] European Committee for Standardization. ISO 50001 Energy Management Systems, 2019.

- [14] Raquel Ponte, Rui Pinto, and Tiago Meireles. Energy Consumption Analysis in SCADA: A Case Study in the Glass Container Industry. In *Conference on Industry Science & Computer Science Innovation (ISCSI 2023)*. Elsevier, 2023. Under submission.
- [15] Department of Engineering, E.S.I., University of Almería. Spain, A. Alcayde, R. Baños, F.G. Montoya, and F.M. Arrabal-Campos. Evaluation of energy consumption and power quality in oil mills using advanced smart meters. *Renewable Energy and Power Quality Journal*, 20:778–782, September 2022. URL: <https://www.icrepq.com/icrepq22/431-22-alcayde.pdf>, doi:10.24084/repqj20.431.
- [16] Álvaro Miguel Almeida Ferreira. Desenvolvimento de uma Aplicação SCADA na Continental – Indústria Têxtil do Ave, S.A. Master’s thesis, University of Porto, 2013.
- [17] Fei Tao, Qinglin Qi, Lihui Wang, and A.Y.C. Nee. Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering*, 5(4):653–661, August 2019. URL: <https://linkinghub.elsevier.com/retrieve/pii/S209580991830612X>, doi:10.1016/j.eng.2019.01.014.
- [18] Ambra Cala, Arndt Luder, Ana Cachada, Flavia Pires, Jose Barbosa, Paulo Leitao, and Michael Gepp. Migration from traditional towards cyber-physical production systems. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pages 1147–152, Emden, July 2017. IEEE. URL: <http://ieeexplore.ieee.org/document/8104935/>, doi:10.1109/INDIN.2017.8104935.
- [19] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. Characterising the Digital Twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 29:36–52, May 2020. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1755581720300110>, doi:10.1016/j.cirpj.2020.02.002.
- [20] Rui Pinto, Gil Gonçalves, Jerker Delsing, and Eduardo Tovar. Enabling data-driven anomaly detection by design in cyber-physical production systems. *Cybersecurity*, 2022. URL: <https://doi.org/10.1186/s42400-022-00114-z>, doi:10.1186/s42400-022-00114-z.
- [21] Eliseu Pereira, Joao Reis, and Gil Goncalves. DINASORE: A Dynamic Intelligent Reconfiguration Tool for Cyber-Physical Production Systems. In *DINASORE: A Dynamic Intelligent Reconfiguration Tool for Cyber-Physical Production Systems*. Eclipse Foundation, September 2020. URL: <https://av.tib.eu/media/51305>, doi:10.5446/51305.
- [22] Manuel Francisco Pinto. Dashboard Development for Energetic Consumption Optimization. Master’s thesis, University of Porto, February 2023.
- [23] SA Circutor. Powerstudio SCADA User Manual (version 4.0.10), 2018.
- [24] SA Circutor. Software Environments for energy management and control, 2016. Catalogue.
- [25] OPC Foundation. What is OPC?, 2023. Last visited on 15-04-2023. URL: <https://opcfoundation.org/about/what-is-opc/>.
- [26] Eugen Diaconescu and Cristian Spirleanu. Communication solution for industrial control applications with multi-agents using OPC servers. In *2012 International Conference on Applied and Theoretical Electricity (ICATE)*, pages 1–6, Craiova, Romania, October 2012. IEEE. URL: <http://ieeexplore.ieee.org/document/6403431/>, doi:10.1109/ICATE.2012.6403431.

- [27] Instrumentation Tools. What is the OPC Server?, 2023. Last visited on 29-06-2023. URL: https://instrumentationtools.com/what-is-the-opc-server/?utm_content=cmp=true.
- [28] Alexandru Stefanov, Chen-Ching Liu, Manimaran Govindarasu, and Shinn-Shyan Wu. SCADA modeling for performance and vulnerability assessment of integrated cyber-physical systems: SCADA MODELING OF INTEGRATED CYBER-PHYSICAL SYSTEMS. *International Transactions on Electrical Energy Systems*, 25(3):498–519, March 2015. URL: <https://onlinelibrary.wiley.com/doi/10.1002/etep.1862>, doi:10.1002/etep.1862.
- [29] Diogo A.C. Narciso and F.G. Martins. Application of machine learning tools for energy efficiency in industry: A review. *Energy Reports*, 6:1181–1199, November 2020. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2352484719308686>, doi:10.1016/j.egyr.2020.04.035.
- [30] Abhijit Dasgupta, Yan V. Sun, Inke R. König, Joan E. Bailey-Wilson, and James D. Malley. Brief review of regression-based and machine learning methods in genetic epidemiology: the Genetic Analysis Workshop 17 experience. *Genetic Epidemiology*, 35(S1):S5–S11, 2011. URL: <https://onlinelibrary.wiley.com/doi/10.1002/gepi.20642>, doi:10.1002/gepi.20642.
- [31] Francisco Rodrigues. Development of Machine Learning Models to Predict glass quality. Master’s thesis, University of Porto, July 2022.
- [32] Ajitesh Kumar. Overfitting & Underfitting in Machine Learning, January 2023. Last visited on 13-04-2023. URL: <https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/>.
- [33] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. URL: <https://doi.org/10.1038/s41586-020-2649-2>, doi:10.1038/s41586-020-2649-2.
- [34] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56–61, 2010. doi:10.25080/Majora-92bf1922-00a.
- [35] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi:10.1109/MCSE.2007.55.
- [36] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro,

- Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.
- [37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [38] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: <https://www.tensorflow.org/>.
- [39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [40] Sebastian Raschka, Joshua Patterson, and Corey Nolet. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, 11(4):193, April 2020. URL: <https://www.mdpi.com/2078-2489/11/4/193>, doi:10.3390/info11040193.
- [41] Great Learning. Hyperparameter Tuning with GridSearchCV, May 2023. Last visited on 12-06-2023. URL: <https://www.mygreatlearning.com/blog/gridsearchcv/>.
- [42] Sinno Jialin Pan. Chapter 21 Transfer Learning. In *DATA CLASSIFICATION: algorithms and applications*. CRC PRESS, S.I., 2020.
- [43] Pavan Kumar Naidu, Shaik Naseer, Shaik Safiya Shaheen, Sai Charan Penugonda, and Dhana Sai Pavan Kumar K. A Review on Smart Energy Meter Based on IOT. *Journal of Algebraic Statistics*, 13(3):1066–1073, 2022.
- [44] Allah Wasaya, Sarib Malik, Muhammad Zaigham Abbas, and Hamza Shahid. An Approach Towards Prepaid Metering System using PowerStudio SCADA. In *2021 4th International Conference on Energy Conservation and Efficiency (ICECE)*, pages 1–5, Lahore, Pakistan, March 2021. IEEE. URL: <https://ieeexplore.ieee.org/document/9406294/>, doi:10.1109/ICECE51984.2021.9406294.

- [45] Sudip Phuyal, Diwakar Bista, Department of Electrical and Electronics Engineering, Kathmandu University Dhulikhel, Jan Izykowski, and Rabindra Bista. Design and Implementation of Cost Efficient SCADA System for Industrial Automation. *International Journal of Engineering and Manufacturing*, 10(2):15–28, April 2020. URL: <http://www.mecs-press.org/ijem/ijem-v10-n2/v10n2-2.html>, doi: 10.5815/ijem.2020.02.02.
- [46] Sebastian Haag and Reiner Anderl. Digital twin – Proof of concept. *Manufacturing Letters*, 15:64–66, January 2018. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2213846318300208>, doi:10.1016/j.mfglet.2018.02.006.
- [47] Thomas H.-J. Uhlemann, Christian Lehmann, and Rolf Steinhilper. The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0. *Procedia CIRP*, 61:335–340, 2017. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2212827116313129>, doi:10.1016/j.procir.2016.11.152.
- [48] Bruno Miguel Rodrigues Martins. *Decision Support System in the Design, Production and Quality Control of Glass Containers*. PhD Thesis, University of Porto, 2017.
- [49] Department of Engineering E.S.I., University of Almería. Spain, A. Alcayde, F.G. Montoya, F.M. Arrabal-Campos, Jesús González, Andrés Ortiz, and R. Baños. Understanding Power Quality using IoT-based Smart Analyzers and Advanced Software Tools. *Renewable Energy and Power Quality Journal*, 19:356–361, September 2021. URL: <https://www.icrepq.com/icrepq21/293-21-alcayde.pdf>, doi:10.24084/repqj19.293.