



# Smart system for monitoring and controlling energy consumption by residential loads

**Paloma Greiciana de Souza Dias**

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering.

Work oriented by:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof. Dr. Luis Cláudio Gambôa Lopes

Prof. Msc. Thadeu Brito

Bragança

2023


# Smart system for monitoring and controlling energy consumption by residential loads

**Paloma Greiciana de Souza Dias**

Dissertation presented to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering.

Work oriented by:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof. Dr. Luis Cláudio Gambôa Lopes

Prof. Msc. Thadeu Brito

Bragança

2023

# Dedication

I dedicate this work to my family and my fiance for helping me get here and become the person I am today.

# Acknowledgement

I would like to thank God for the opportunity to carry out this project and successfully conclude my objectives.

To my mother Greiciana and my grandmother Zilda who always supported and encouraged me in my studies. Thanks to these two women, I got where I am now.

To my sisters Pâmela and Camille and my brother Vitor who always gave me support and inspired me to do my best. They are my base and my pride.

To my fiancé Mateus who was by my side at all times, who always made me see my potential and with whom I could share incredible and challenging moments.

To my uncles Joaquim and Edivaldo, my aunt Maria José, and my cousin Higor for always encouraging and supporting me.

To Mrs. Lourdes and Mr. José Luiz who have accompanied much of the process to get here and have given their full support.

To my friend Samuel who was by my side in the worst and best moments of life and believed in me.

To my parents-in-law Sandra and Geraldo and my fiancé's family who from the beginning showed their support and care for me.

To my brother-in-law Junior for his partnership and friendship.

To my supervisors José Lima, Luis Gambôa and Thadeu Brito for their constant guidance, technical knowledge and evaluations.

To the technicians of the laboratory where I worked, Mr. Batista and Filipe, for the support with tools, equipment and also knowledge.

To CEFET-MG and IPB that gave me the opportunity to get a double degree and

also the orientation to do the whole master's process.

To my friends Ana Carolina, Lucas Melo, Vinicius Ferreira and William who encouraged me countless times, cared for and believed in me, even when I didn't believe in myself.

To my friends Luíza, Joana, Karine, Juliana, Lucas Guimarães, Leandro, Jefferson, Vinicius Veridiano, Daniela, Daniel, João Marcos, Karen, Thaís, and Yago at CEFET-MG who inspired me and learned a lot. I will always carry these people in my heart.

To my friends Vitor, Renata, Débora, João Vitor, Nathália, Júlia, Maria, Andréia, Thaís, Jefferson, Ana Letícia, Mariana, Ana Gabriela, Danilo, Igor, Leonardo, Lara, Zsuzsanna, Afag, Altynay and Bianca from the exchange program who were incredible with me throughout the whole process and with whom I lived wonderful moments. I am very grateful to have met these special people.

# Abstract

This work proposes the development of a smart system for monitoring and controlling surplus energy consumption by residential loads connected to smart plugs. Data processing, storage and monitoring tools will be used, specifically Node-RED, InfluxDB, Grafana and Home Assistant. The latter allows remote control of devices through its interface. Through these tools, the user can visualize excess power data, which, in this work, are obtained through a hardware prototype that uses the ESP32 as a microcontroller and is responsible for measuring voltage and current of energy to analyze negative power and positive or surplus. Based on this measurement, load control tests were carried out based on the ratio of available energy to the power required by the load, using the Home Assistant interface. Then, this work focused on performing the excess power prediction by the linear regression method, a Machine Learning approach. A dataset with values of consumption and generation of photovoltaic energy in a residence was used and forecast analyzes were performed. The predicted results meet the expected objectives, although it is reasonable to conclude that further studies can still be done to ensure the robustness of the prediction model.

**Keywords:** Surplus energy; smart plug; Machine Learning

# Resumo

Este trabalho propõe o desenvolvimento de um sistema inteligente de monitorização e controlo de consumo de energia excedente por cargas residencias ligadas a smart plugs. Serão utilizadas ferramentas de processamento, armazenamento e monitorização de dados, especificamente Node-RED, InfluxDB, Grafana e Home Assistant. Esta última permite o controlo remoto de dispositivos por meio de sua interface. Através dessas ferramentas, o utilizador pode visualizar dados de potência excedente, que, nesse trabalho, são obtidos por meio de um protótipo de hardware que utiliza o ESP32 como microcontrolador e é responsável pela medição de tensão e corrente de energia para analisar potência negativa e positiva ou excedente. Com base nessa medição, foram feitos testes de controle de carga a partir da relação da energia disponível com a potência exigida pela carga, utilizando a interface do Home Assistant. Em seguida, esse trabalho focou em realizar a previsão de potência excedente pelo método de regressão linear, uma abordagem de Machine Learning. Foi utilizado um dataset com valores de consumo e geração de energia fotovoltaica em uma residência e análises de previsão foram realizadas. Os resultados previstos atendem aos objetivos esperados, embora seja razoável concluir que novos estudos ainda podem ser feitos para garantir a robustez do modelo de previsão

**Palavras-chave:** Energia excedente; smart plug; Machine Learning

# Contents

**Acknowledgement** **vii**

**Abstract** **ix**

**Resumo** **xi**

**Acronyms** **xxiii**

**1 Introduction** **1**

1.1 Framework . . . . . 2

1.1.1 Power loss in the distribution network . . . . . 2

1.1.2 Types of consumers most common in Portugal . . . . . 4

1.1.3 Influence of distributed generation on the grid . . . . . 6

1.1.4 Costs of selling surplus energy to the grid . . . . . 6

1.1.5 Cost of batteries . . . . . 8

1.2 Proposed use of surplus energy . . . . . 9

1.3 Objectives . . . . . 11

1.4 Work Structure . . . . . 12

**2 State of art and Study of tools** **15**

2.1 Data visualization in the Internet of Things . . . . . 15

2.2 Monitoring and Control of Energy Consumption . . . . . 16

2.3 Priority of electrical devices . . . . . 19

2.4 Machine Learning in residential energy management . . . . . 20
2.4.1 Concept of Machine Learning . . . . . 21
2.4.2 Machine Learning case studies . . . . . 21
2.5 Used tools . . . . . 22
2.5.1 ESP32 . . . . . 22
2.5.2 Message Queuing Telemetry Transport (MQTT) Protocol . . . . . 23
2.5.3 Node-RED . . . . . 24
2.5.4 InfluxDB . . . . . 24
2.5.5 Grafana . . . . . 25
2.5.6 Home Assistant . . . . . 25

**3 Development** **27**
3.1 System architecture . . . . . 27
3.2 Machine Learning . . . . . 29
3.3 Monitoring and control system configuration . . . . . 31
3.3.1 Initial control tests . . . . . 32
3.3.2 Surplus power monitoring . . . . . 32
3.4 Machine Learning: Application of linear regression . . . . . 40
3.4.1 Pre-analysis of the dataset . . . . . 40
3.4.2 Application of Linear Regression . . . . . 43
3.4.3 Forecast with hours . . . . . 45
3.4.4 Forecasts from one day to the next . . . . . 46
3.4.5 Using 5 days for forecast . . . . . 46
3.4.6 Using 10 days for forecast . . . . . 47
3.4.7 Using 30 days for forecast . . . . . 48

**4 Results** **51**
4.1 Surplus power monitoring . . . . . 51
4.2 Analysis of Machine Learning . . . . . 57
4.2.1 Forecast with hours . . . . . 58

4.2.2 Forecasts from one day to the next . . . . . 65
4.2.3 Using 5 days for forecast . . . . . 69
4.2.4 Using 10 days for forecast . . . . . 73
4.2.5 Using 30 days for forecast . . . . . 76

**5 Conclusion and Future Work** **79**

**A Original Project Proposal** **93**

**B C programming code** **96**

**C Data for the first analysis of Linear Regression** **108**

**D Python code for Linear Regression** **113**

**E Prediction results with models created from hours** **120**

# List of Tables

4.1 Model created on May 14 / Forecast for May 15 . . . . . 66
4.2 Model created on May 29 / Forecast for May 30 . . . . . 67
4.3 Model created on June 3 / Forecast for June 4 . . . . . 68
4.4 Model created on June 13 / Forecast for June 14 . . . . . 69
4.5 Model created on May 14 - 18 / Forecast for May 19 . . . . . 70
4.6 Model created on May 14 - 23 / Forecast for May 24 . . . . . 74
4.7 Model created on May 14 - June 13 / Forecast for June 14 . . . . . 77

# List of Figures

1.1 Data on energy losses in the distribution network in Portugal, between 1999 and 2019 [9]. . . . . 3
1.2 Number of electricity consumers by type of consumption in Portugal, between 1994 and 2021 [13]. . . . . 4
1.3 Electricity consumption by type of consumption between the years 1994 and 2021 in Portugal. . . . . 5

3.1 Schematic of the proposed system. . . . . 29
3.2 Command processing by Node-RED for LED turning on and off. . . . . 32
3.3 Grid voltage reading. . . . . 33
3.4 Reading the input voltage on the microcontroller. . . . . 34
3.5 Construction of the prototype for simulating excess power using variable resistance. . . . . 35
3.6 Building processing of voltage and current data and InfluxDB storage by Node-RED. . . . . 35
3.7 Building processing of power data and InfluxDB storage by Node-RED. . . 36
3.8 Simplified example of an interface in Home Assistant that allows the user to enter the load power. . . . . 36
3.9 Processing of the load power data by Node-RED. . . . . 37
3.10 Prototype for current and voltage measurement connected to a photovoltaic inverter. . . . . 39

3.11 Relation between photovoltaic generation and energy consumption by circulation pump, dishwasher, freezer and washing machine in the residence under analysis. . . . . 42
3.12 Relation between surplus energy and energy consumption by circulation pump, dishwasher, freezer and washing machine in the residence under analysis. . . . . 42
3.13 Relation between Y and X values used to create forecast models with May 14th-18th. . . . . 47
3.14 Relation between Y and X values used to create forecast models with May 14th-23th. . . . . 48
3.15 Relation between Y and X values used to create forecast models with May 14th - June 13th. . . . . 49

4.1 RMS voltage data measured by prototype and stored in InfluxDB. . . . . 52
4.2 RMS current data measured by prototype and stored in InfluxDB. . . . . 52
4.3 Simulated surplus power data measured by prototype and stored in InfluxDB. 52
4.4 Simulated surplus power data measured by prototype and viewed at Grafana. 53
4.5 Simulated surplus power data measured by prototype and viewed at Home Assistant. . . . . 53
4.6 Graphical representation of the voltage, current, and power values at a load connected to the grid. . . . . 54
4.7 Graphical representation of the electric current in the load supplied by the electrical network. . . . . 54
4.8 Graphical representation of the voltage, current, and power values in a photovoltaic inverter. . . . . 55
4.9 Graphical representation of the electric current inserted into the grid in a photovoltaic inverter. . . . . 56
4.10 Power data read from PV inverter and stored in InfluxDB. . . . . 56
4.11 Power data read from PV inverter and viewed at Grafana. . . . . 57

4.12 Power data read from PV inverter and viewed at Home Assistant interface. 57
4.13 Graphical representation of forecasts using 1 hour of day 14 to forecast the rest of the day. Due to the proximity of the values obtained with the models that use 1 to 3 features as X predictor values, the graphical visualization of all the results was impaired. . . . . 58
4.14 Mean absolute error in forecasts with models created from hours data on May 14th. . . . . 60
4.15 Mean absolute error in forecasts with models created from hours data on May 21st. . . . . 60
4.16 Mean absolute error in forecasts with models created from hours data on May 28th. . . . . 61
4.17 Mean absolute error in forecasts with models created from hours data on June 4th. . . . . 62
4.18 Accuracy of the regression models created from the first 12 hours of May 14. 62
4.19 Mean absolute error and median absolute error values considering forecasts for the 31 days following May 14th. . . . . 63
4.20 Error variance values considering forecasts for the 31 days following May 14. 64
4.21 Explained Variance Score values considering forecasts for the 31 days following May 14. . . . . 64
4.22 Forecast for May 15 based on May 14 data. . . . . 65
4.23 Forecast for May 30 based on May 29 data. . . . . 66
4.24 Forecast for June 4th based on June 3 data. Due to the proximity of the values obtained with the models that use 1 to 6 features as X predictor values, the graphical visualization of all the results was impaired. . . . . 67
4.25 Forecast for June 14 based on June 13 data. . . . . 68
4.26 Forecast for May 19 based on May 14 - 18 data. . . . . 70
4.27 Mean absolute error and median absolute error values considering forecasts for the same day 19th in 11 months based on May 14th-18th data. . . . . 71

4.28 Error variance values considering forecasts for the same day 19th in 11 months based on May 14th-18th data. . . . . 72
4.29 Explained Variance Score values considering forecasts for the same day 19th in 11 months based on May 14th-18th data. . . . . 72
4.30 Forecast for May 24 based on May 14 - 23 data. . . . . 73
4.31 Mean absolute error and median absolute error values considering forecasts for the same day 24th in 11 months based on May 14th-23rd data. . . . . 74
4.32 Error variance values considering forecasts for the same day 24th in 11 months based on May 14th-23rd data. . . . . 75
4.33 Explained Variance Score values considering forecasts for the same day 24th in 11 months based on May 14th-23rd data. . . . . 76
4.34 Forecast for June 14 based on May 14 - June 13 data. . . . . 77
4.35 Linear Fit of the linear regression model using 1 feature. . . . . 78

E.1 Graphical representation of forecasts using 6 hours of day 14 to forecast the rest of the day. Due to the proximity of the values obtained with the models that use 1 to 6 features as X predictor values, the graphical visualization of all the results was impaired. . . . . 121
E.2 Graphical representation of forecasts using 6 hours of day 14 to forecast the day 15. Due to the proximity of the values obtained with the models that use 1 to 6 features as X predictor values, the graphical visualization of all the results was impaired. . . . . 122
E.3 Graphical representation of forecasts using 12 hours of day 14 to forecast the rest of the day. . . . . 123
E.4 Graphical representation of forecasts using 12 hours of day 14 to forecast day 15. . . . . 124

# Acronyms

**ADR** Automated Demand Response.

**BI** Business Intelligence.

**DPA** Personal Digital Assistants.

**DR** Demand Response.

**EMS** Energy Management System.

**ERSE** *Entidade Reguladora de Serviços Energéticos*.

**ESTiG** Escola Superior de Tecnologia e Gestão.

**GUI** Graphical User Interface.

**HC** Hosting Capacity.

**HEM** Home Energy Management.

**HEMS** Home Energy Management Systems.

**HMI** Human-Machine Interface.

**IBM** International Business Machines Corporation.

**IBR** Inclined Blocking Rate.

**IDE** Integrated Development Environment.

**IoT** Internet of Things.

**IPB** Instituto Politécnico de Bragança.

**M2M** Machine-to-Machine.

**MAE** Mean Absolute Error.

**MdAE** Median Absolute Error.

**MQTT** Message Queuing Telemetry Transport.

**NILM** Non-Intrusive Load Monitoring.

**P.I.R** Passive Infrared.

**PV** Photovoltaic.

**REN** *Redes Energéticas Nacionais.*

**RTC** Real-Time Clock.

**SCADA** Supervisory Control And Data Acquisition Systems.

**TOU** Time-of-Use.

**TSDB** Time Series Database.

**UPAC** *Unidades de Produção para Autoconsumo.*

**UPP** *Unidades de Pequena Produção.*

**VOLL** Value of Lost Load.

# Chapter 1

# Introduction

Smart energy management systems, such as smart grids and smart meters, have become a focus of technological development within the energy sector. Implementing these systems includes using the Internet of Things (IoT) to ensure the connection between parts of the communication network and the safe transport of data [1]. Within the context of smart homes, consumers can remotely monitor and control household devices, which enables energy savings [2] and conscious electricity consumption. Devices such as smart plugs can connect ordinary plugs to appliances with functions to manage energy consumption and turn the devices on/off based on this monitoring. Control can be done via an Internet connection using an application, which can have features such as viewing the energy consumption of appliances and scheduling the time at which the appliances will be turned on or off [3].

The residential energy management scenario is characterized by changing load usage without compromising consumer comfort [4]. In addition, topics such as Home Energy Management Systems (HEMS) have been popular because they enable efficient energy management, reducing consumption and causing bill savings as well as decreasing the use of fossil fuels. HEMS allows the intelligent use of appliances, which includes controlling the operation of loads, also according to consumer preferences. In this way, these systems facilitate efficient energy use and consumer comfort simultaneously. In this context, the HEMS architecture can include distributed generation. Thus, the meter devices provide

the system with consumption and generation data [5].

In the case of distributed generation, the consumer unit generates electric energy through renewable sources. These systems can be connected directly to microgrids or to the main grids. In [6], it mentions the use of batteries to store surplus energy in a microgrid and supply the consumer units when needed. However, considering the high costs involved in purchasing a battery bank, another way to save the surplus energy is to supply it to the grid. Although these approaches treat the issue of surplus power generation, this work proposes a new technique in this field: developing a smart system capable of directing surplus power to specific loads in the home based on the power supply and consumption pattern. This work also presents smart plugs which provide the necessary data to the system and control the energy consumption of the electrical devices.

## 1.1 Framework

The subject of this work involves the development of a system to monitor and control the power consumption of devices connected to smart plugs and the surplus energy, remotely, in cases where consumers produce their own power. Users can do the ON/OFF control of the devices or control the power at which specific loads operate. However, the main focus of this work is to develop a system capable of predicting surplus energy and consumption, enabling the future decision-making capability of the system to be applied to the use of this energy by loads. In this way, this topic will consider the current context of energy generation and consumption, as well as the reasons behind the objectives of the work. The following subtopics will provide the basis for the general and specific objectives aimed by the project.

### 1.1.1 Power loss in the distribution network

Power generation systems use transmission and distribution networks to transport and supply electricity to consumers. Transmission networks include high, extra-high, and ultra-high voltage lines, which are connected to sub-transmission networks, also considered

by certain utilities to be part of the distribution network, with voltages from 230kV up to 35kV; and below 35kV is the distribution network [7]. This has the function of directly supplying the demand. However, most of the power losses occur in the distribution network [8].

In Portugal, the *Entidade Reguladora de Serviços Energéticos (ERSE)* published information on distribution network losses between 1999 and 2019. Figure 1.1 was adapted from [9] and presents the mentioned data graphically.

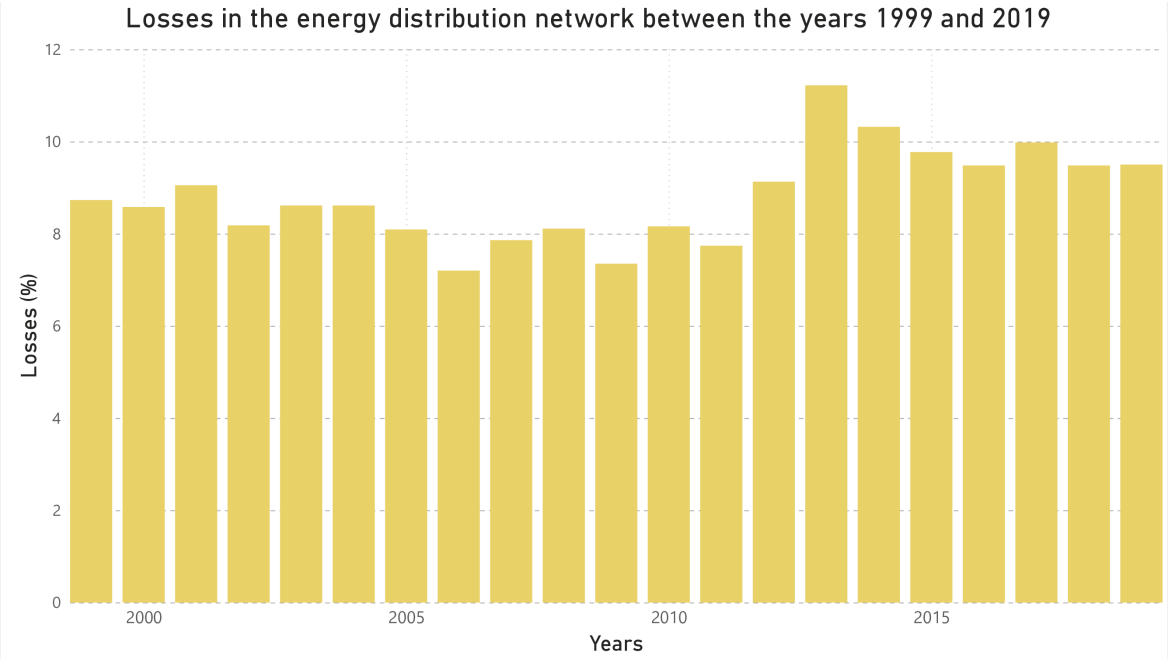


Figure 1.1: Data on energy losses in the distribution network in Portugal, between 1999 and 2019 [9].

As can be seen in the graph of Figure 1.1, the losses between the years 1999 and 2019 varied in a range between 7% and 11%, approximately. It is worth noting that the years after 2012 were characterized by losses of more than 9%, with 2013 being the year with the highest losses, above 11%.

Regarding the cost of losses, the electricity tax system in Portugal has an adjustment factor for losses that adjust according to the voltage level and the time period [10]. In this way, according to the 2021 Tariff Regulation for the electricity sector [11], of the ERSE,

tariff values are converted to the voltage level and tariff mode of consumers supplied by the last resource suppliers, taking into account loss adjustment factors. Thus, it is possible to highlight that the costs involved in energy losses in the network are distributed to consumers.

### 1.1.2 Types of consumers most common in Portugal

Energy consumption in Portugal is mainly composed of low-voltage consumers, according to the ERSE [12]. In this same context and to the present moment, PORDATA, the official statistics site of Portugal, presents data until 2021 about electricity consumers by type of consumption. Figure 1.2 presents data taken from this site, in a way adapted from [13].

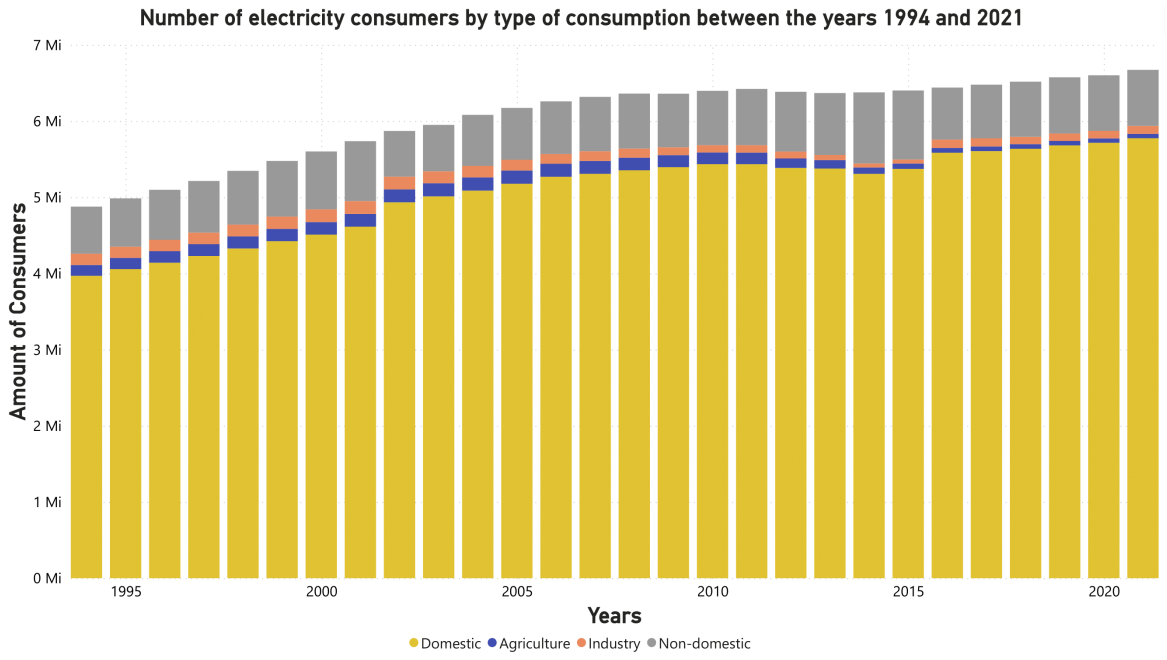


Figure 1.2: Number of electricity consumers by type of consumption in Portugal, between 1994 and 2021 [13].

As can be seen in the graph in Figure 1.2, in all years between 1994 and 2021, the domestic class is responsible for most of the electricity demand in Portugal. In this

context, Figure 1.3 presents consumption in kWh by type of consumption in Portugal, including the domestic class. The image was adapted from [14].

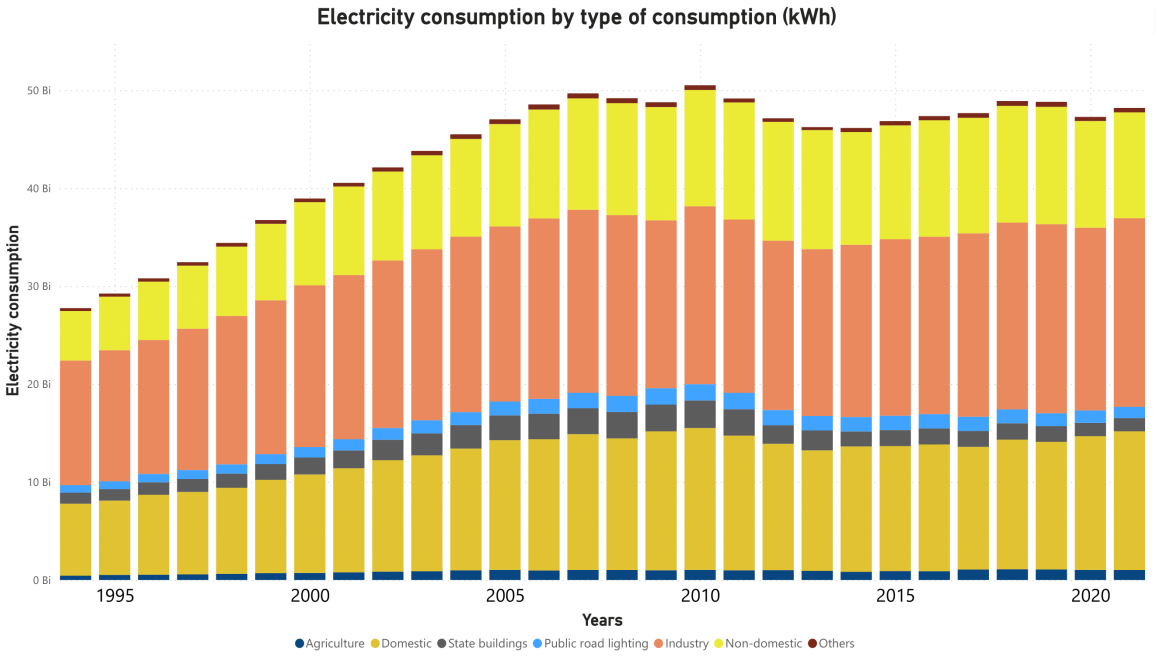


Figure 1.3: Electricity consumption by type of consumption between the years 1994 and 2021 in Portugal.

It can be seen in the two graphs previously presented that domestic consumers represent approximately 86% of consumers in Portugal. This group consumes about 30% of the electricity in the country. Although this number is not the majority of the electricity consumed in the country, this analysis was made for this work in view of the application of the smart monitoring and control system, especially in residential contexts. Understanding that the domestic public is the widest among electricity consumers, the scalability of the system will indeed be useful and extensive. However, considering that these consumers do not use most of the country's electricity, it is interesting to expand the application of the system to other consumption sectors such as industry, for example.

### 1.1.3 Influence of distributed generation on the grid

The structure of the distribution network needs to meet the minimum requirements of security, stability, and quality. In this way, it must be adapted to the different loads and operating characteristics of the consumers [15]. Distributed generation is one of the elements that can directly affect this infrastructure both in a positive way, from the mitigation of peak currents with respect to the operation of loads such as air conditioners, and in a negative way, such as fault currents, voltage and power fluctuations, changes in power factor, reverse power flow, among others [15].

In [16], the problem of high insertion of distributed generation in the electrical grid was discussed, which includes collateral effects such as over and under voltages, overloads, protection failures, and other operational damages in the grid. The authors focused on presenting analysis, research, and techniques for evolving the Hosting Capacity (HC) of the grid, which provides improvements in system security, reliability, and flexibility.

In this scenario of high insertion of distributed generation in the grid, and despite the evolution of research and technologies for the development and preparation of the distribution grid infrastructure regarding the connection of distributed generation, the measures taken by the proposed smart system redirect the surplus energy generated for load demand, in order to satisfactorily and efficiently supply the needs of *prosumers*, a term used to refer to consumers who are also energy producers [17]. Besides satisfying the expectations of the users, the system mitigates the insertion of surplus energy in the electric grid, reducing the impacts of the high penetration of distributed generation.

### 1.1.4 Costs of selling surplus energy to the grid

To understand the case of selling surplus energy to the energy market, information that deals with the distributed generation with a renewable source, specifically solar energy will be cited. In Portugal, in the context of selling surplus photovoltaic generation, two types of units should be considered: the *Unidades de Pequena Produção* (UPP), which have a instalation less or equal to 250 kW with the grid and whose production is entirely

intended for export to the grid; and the *Unidades de Produção para Autoconsumo* (UPAC), which produce energy for self-consumption, and may or may not be connected to the grid [18]. The payment to producers is 0.045 €/kW [19]. However, this topic will focus on the UPACs, since the producers in these units consume the energy generated. The surplus generated in this case can be supplied to the grid free of charge or it can be sold on the energy market, provided that it satisfies the requirements established by the regulatory authorities [20].

In self-consumption units, the selling of surplus generation can be determined through contracts between producers and market agents, with previously established conditions and respecting the regulations of the ERSE [21]. To exemplify, in [22] the authors worked on the evaluation of the economic viability of a photovoltaic installation in the Campus Politécnico de Viseu, in Portugal. Regarding the surplus production of this UPAC, the authors adopted as reference the value of 0.0504€/kWh, established by *Redes Energéticas Nacionais* (REN), in its 2015 annual report on the electricity market [23]. However, the 10% correction was applied as mentioned in [24], which led to a value of 0.045€/kWh as the selling value for the energy injected into the grid.

Within the requirements necessary for the selling of surplus energy, the prosumer needs to have a bidirectional meter that will be responsible for measuring the energy exported and imported from the grid [25]. The energy injected into the grid is measured within 15 minutes [26]. It should be taken into account that the consumer needs to be properly registered and authorized to exercise the function of electricity seller [21].

Thus, especially in the context of photovoltaic generation, it is worth noting that the surplus selling values by a consumer unit is not relevant when compared to the electricity tariff values, because they are equivalent to only 10% of the energy that is purchased from the grid by the users [27].

### 1.1.5 Cost of batteries

Another way to save the surplus energy is through the use of batteries. Considering the advancement of renewable energy as well as the use of electric vehicles, battery systems are evolving with time [28]. In [29], it can be seen that the use of energy storage in batteries in a photovoltaic generation system is able to significantly reduce annual energy bills. However, the authors analyze the lack of profitability in an investment made in a battery bank, since analyzing a system whose operation is defined in 365 cycles per year and a maximum discharge capacity of 70%, in a battery of 4200 cycles, a lifetime of 10.5 years was estimated, which would require a cost of 205€/kWh in a power generation project to ensure the profitability of the system. Regarding the levelized cost of energy storage, [30] presents the possibility of reduced storage costs for 2023.

Regarding Lithium-ion batteries, in [31], it is mentioned that with a cost of €200/kWh and a lifetime of 2500 cycles or more, these devices would become more attractive in grid-connected system applications. Li-ion batteries are the most chosen by the industry for their advantageous characteristics compared to other batteries, such as higher efficiency, power and energy density, and their price has reduced dramatically after 2013, costing around US$156/kWh in 2019 [32]. A cost projection for this type of battery with 4-hour operation duration was made in [33], which presented results of equivalent prices of US$143/kWh, US$198/kWh and US$248/kWh as low, medium and high value respectively in 2030, and US$87/kWh, US$149/kWh and US$248/kWh in 2050.

Despite the evolution of the use of batteries for energy storage, it can be stated that the cost involved in the purchase and installation of this equipment is still expensive. In [34], although the author mentions different applications of battery systems, the high price paid for these devices is undeniable. In [35], it is also mentioned that the cost of lithium-ion batteries in the market is high and residential consumers resist buying these devices for this reason. Thus, the proposal of this work is of high importance, since it provides a new direction for the use of surplus energy in the detriment of existing modes, ensuring the comfort of the user and enabling the efficient use of this energy.

## 1.2 Proposed use of surplus energy

To begin a coherent analysis of surplus power applications outside of the previously mentioned contexts, scenarios of large surplus generation, at industrial and urban levels, will be considered. After that, the focus will be given to residential applications. In [36], it is mentioned that in Sweden in 1980 the use of surplus power generated in nuclear power plants was carried out by directing this energy to supply electric resistances in domestic heating, electric boilers in heat pumps in single-family houses, electric boilers for industries, and electric boilers and pumps for district heating. The last is the topic on which the mentioned work emphasizes. Especially in contexts of evolving power generation through solar, wind, and hydro sources, where surplus production is reasonably predictable, district heating systems can benefit from this surplus as well as alleviate problems of imbalance in the power grid.

Advancing the analysis of the application of surplus energy to the rural environment, one can mention economic activities in which the application of the proposed system would be feasible. In [37], although no focus is given to the use of surplus energy, the authors deal with the use of the photovoltaic system as a power supply for equipment used in rural applications, such as water pumps for irrigation, water pumping and aerators for fish farming, air conditioning in aviaries, electrification of fences and desalination. It is worth emphasizing that the photovoltaic system is not the only means of a generation that makes it possible to obtain surplus energy. Another detail is that in loads such as water pumps, for example, a preparation of the smart surplus power management system should be made to ensure the proper operation of the pump respecting electrical parameters, and operation time, among others, without compromising its operation.

Another example of using surplus energy in rural applications is in the residential context. In [38], a procedure is observed for evaluating a photovoltaic system with the use of batteries in order to use the surplus energy to meet the demand for water pumping in a rural residence in Malaysia. The load demand of the residence equals 3.2 kWh/day, the battery power sizing was set at 3.55 kWh and the water pump operation is $363 m^3$/day

with 6 meter elevation. It can be seen that there is no way to avoid the generation of excess energy, which implies the great relevance of applications that use this excess in their operations, such as water pumping and irrigation.

In residential scenarios, one application mentioned in [39] is water heating. The authors have developed a system for using surplus photovoltaic energy for heating water either by means of electric storage heaters or by means of heat pumps. Without appropriate control, it is understood that such a system is not feasible because the heating demand at a given time may not match the surplus available at that moment, which does not result in large savings in energy use. Therefore, a control was made in this system to guarantee the conciliation between the demand of the water heating load and the available energy, and a storage tank was used to guarantee thermal stability. In reporting on the benefits resulting from this work, the authors report the savings in the use of grid energy for water heating and the reduction of sending surplus energy to the grid, which avoids unbalancing factors due to the high insertion of distributed generation.

In addition to the applications already mentioned, surplus energy can be directed to the generation of hydrogen. In [40], it is presented that natural gas has a higher hydrogen-to-carbon ratio among the forms of hydrogen generation, making it a significant compound for hydrogen production. However, it is the use of fossil fuels, which leads to CO2 emissions. Electrolysis is a more expensive process than natural gas reforming. About 80% of its cost is for the electricity required in its process. Therefore, it is possible to couple the production of hydrogen by electrolysis with a renewable energy source. In addition, one of the products of electrolysis is oxygen gas, which can be used primarily in industrial applications. But only 5% of current hydrogen production is based on water electrolysis.

In [41], a study was done on surplus energy management in a microgrid connected to renewable energy generation systems (solar and wind). Three independent situations were discussed: storage, the use of the energy for hydrogen generation with storage capacity, and the insertion of the surplus into the electrical grid. Focusing on hydrogen generation, the authors mention that its production occurs by means of an electrolyzer and it can

be used as an energy source to power system loads or the grid itself. Thus, producing hydrogen by means of the excess energy supplied by other sources is a viable application and can mitigate the effects of the surplus on the electric grid.

Thus, it is possible to observe that there are several applications in which it is possible to use the surplus energy generated, avoiding expenses with batteries or insertions in the electrical grid. In this sense, the focus of this work is on the residential context and countless loads can make use of this excess energy, as long as the control and management system is properly in accordance with the electrical and operation parameters of the loads, besides the information related to the amount of energy available and the consumption pattern of the user.

## 1.3 Objectives

The objective of the proposed work is defined by the development of a system for monitoring and controlling the consumption of surplus energy generated by residential loads connected to smart plugs. For this control to be possible, it is necessary that the system has access to the excess generation and consumption information of the user, which is provided by the plugs. This process will provide data that will define excess energy generation and consumption history. Based on this, the system will be able to predict these variables and automatically make decisions regarding the direction of excess energy to specific loads in the home.

In addition, it should be considered that the control actions for the electrical devices will be sent to the smart plugs. This control can also be done manually because by monitoring the surplus and consumption, the user can decide which load to switch on or off. One of the differentials of the smart plugs considered is the ability to control not only the ON/OFF of the loads, but also their operating power, based on the load they are connected to. The focus of this work does not involve how to perform this control, but the prediction strategies used for automatic decision-making.

For monitoring, the consumer will have at his disposal an interface from which he/she

will visualize parameters such as the surplus energy generated and consumption. Through the same interface, the user can control the operation of loads, based on the monitored information. The consumption and surplus energy data are also stored in a database and will be used by the system to create a forecast model, which will enable automatic decision-making regarding the operation of the loads. It is intended to use Machine Learning tools to achieve this goal [42]. It should be noted that control actions are also stored in the database.

In this sense, the following presents the specific objectives of this work based on the details of the development of the project:

- Configure Node-RED as a data processing tool, InfluxDB for information storage, Grafana for monitoring and Home Assistant for data visualization and device control.
- Develop a hardware prototype to measure voltage, current and power and thus identify surplus energy generation.
- Perform tests to control devices in a manual way.
- Perform data analysis and application of prediction models.

With the background presented and taking into consideration the relevance of the topic, this work will present the development of the project and the detailed steps to achieve the objectives of the proposed system.

## 1.4 Work Structure

The work is organized as follows: in Chapter 2 the state of the art is described, which presents works related to the presented objectives and definitions of important tools and concepts; in Chapter 3 the methodology used for the execution of the project is presented, which includes approaches on the performance of the adopted tools, illustrative schemes on the functioning of the system as well as the tasks performed by it. In addition, Chapter

3 also presents the all stages of system development are described in detail; in Chapter 4 the results achieved with the execution of the stages objectified in the work are displayed; finally, Chapter 5 presents the conclusion of the work, which includes analyzes of what was developed and possibilities for improvements in the future.

# Chapter 2

# State of art and Study of tools

This chapter presents works that approach concepts inserted in the scenario of monitoring and control of intelligent systems. Section 2.1 presents a literature review about data visualization in the context of the early IoT. Then, Section 2.2 deals with approaches related to energy consumption and energy monitoring control systems and also shows applications of IoT systems and Big Data. Section 2.3 shows the application of electrical devices' priority in the context of energy consumption control. Section 2.4 presents definitions and applications of Machine Learning in residential energy management. Finally, in Section 2.5 the software tools that were used for this work and the reason for their choice compared to other platforms are presented.

## 2.1 Data visualization in the Internet of Things

Nowadays is defined by the great development of technologies linked to the IoT and smart systems, in addition to the constant need to monitor, analyze and process a large volume of data. In this context, platforms for viewing and monitoring the behavior of systems are included. In [43], Carro developed a system for monitoring and controlling environmental conditions, using IoT technologies. For this, sensors acquire data related to temperature, humidity, brightness, presence, opening and closing of doors and windows, the concentration of volatile compounds and CO2 in the air, and electrical consumption. These

data were stored in the InfluxDB database and visualized with the help of the Grafana platform, which made it possible to take measures consistent with energy efficiency. The system provides alerts to the user, allowing him to take action in relation to environmental conditions.

In [44], Queiroz et al. worked with the development of IoT and Big Data concepts by carrying out a testbed with a DC motor, whose read information that was monitored by an intelligent system in real-time. Platforms such as Node-RED were used for this application, in which flows were created that retrieved and integrated the data captured from the engine and stored them in a MongoDB database.

## 2.2 Monitoring and Control of Energy Consumption

Currently, the monitoring of energy consumption is of vital importance for taking measures that lead to a more conscious use of energy, the reduction of the exacerbated use of energy resources, and the consequent development of energy efficiency. In the context of energy consumption monitoring and control systems, in [45], Hernández et al. worked on a surplus energy management system for distributed generation. The software used to carry out the monitoring was LabView® 8.5, from which an interface was built that allowed the visualization of the system parameters. This could be operated in manual mode, which gave the user the possibility to interact with the system, deciding whether to work as a grid-connected photovoltaic system or an autonomous system using batteries.

Still within the reality that involves energy management and introducing the concept of the IoT in the energy sector, the great evolution of intelligent systems is remarkable, not only on the distribution side, but also on the demand side. As an example, in [46], Al-Ali et al. published a paper on an Energy Management System (EMS) for smart homes, in which they used IoT and data analytics platforms. In this context, data regarding energy consumption and environmental conditions were collected by the microcontroller and sent to a server. Then, the information was stored in a database and analyzed using Business Intelligence (BI) tools, so that it could later be organized in a visually

understandable way. This format was defined by tables, graphs and reports, which were presented to energy consumers through a mobile platform. The system also gave the user the ability to remotely control home devices. Basically, the consumer could, through the mobile application, turn on/off devices, which were connected to relays commanded by the microcontroller.

Within a broader context that involves consumer and utility, in [47], Hussain et al. developed a load monitoring and control system using LoRa technology and Supervisory Control And Data Acquisition Systems (SCADA) server. The project consisted of a load energy consumption measurement system, whose data were processed and sent to a gateway via LoRa. From the gateway and using the Message Queuing Telemetry Transport (MQTT) communication protocol, the information was sent to a SCADA server managed by the power utility. This server had a data visualization interface and functions such as alarms, and report development, among others. The information was analyzed in real-time and the SCADA system allowed ON/OFF control of the loads.

In addition to this topic, in [48], Silva developed a system for monitoring energy consumption by electrical loads. For this, specific loads were used from which electrical quantities were measured, such as active, reactive, and apparent power, effective current and voltage, and the power factor. For data processing, the ESP32 microcontroller was used, due to its wide applicability and relevant resources, such as Wi-Fi communication and connection to the Arduino IDE. From there, the information was sent to the InfluxDB database, a type of Time Series Database (TSDB) and opensource database, commonly used in applications that deal with real-time data. To visualize the measured quantities, Grafana was used as a tool, which provides the user with a real-time view of the measurement system, as well as histories and alerts.

Furthermore, in [49], Carvalho developed a model that addresses the behavior of loads intended for water heating and space heating in a smart home with a wind generator in order to reduce electricity consumption without interfering with comfort of the user. The developed algorithm also aims to minimize the values of the energy bill based on the tariff analysis. It is worth mentioning that the possibility of consumer interaction with the

smart home system is approached from an interface in which the user defines the moment when the space heater and water heater will work.

In addition to this topic, in [50] Muliadi et al. carried out real-time monitoring of a residence's energy consumption. Sensors read specific energy consumption data and send it to the microcontroller, which processes the information and sends it to a web server. Such data were stored in a database. The web server was built using the Sublime Text text editor and the PHP language, and the access to the database server was made using the XAMPP application. The stored data can be accessed by a web application, which was developed using Sublime Text tools as a text editor, as well as HTML and CSS languages for the frontend and PHP and MySQL for the backend.

Still in the energy consumption monitoring and control scenario, in [51], Chaudhari et al. They developed a smart plug using the ACS712 current sensor, which sent the data to the Arduino, from where it was sent to the Raspberry Pi. Then this information was stored in a database. Thus, the user can track consumption through a website and Android application, which the consumer accesses with login. In this context, the project generates graphs that show the energy consumed by the devices and calculate the electricity bill. In addition, the objective of the work is also to give the user the possibility to control devices remotely and visualize when the energy consumption is higher than expected. Therefore, this work contributes to the conscious consumption of energy at a low cost.

Another relevant work in smart energy management is described in [52]. The authors considered an intelligent house for the application of an efficient energy management system, in the context of the presence of solar power generation, the possibility of storing the surplus in batteries, and the presence of lighting loads, air conditioners, and other electrical devices. Among the approaches of the strategy used by the authors is to charge the battery at times when there is surplus energy and discharge it at nighttime hours. After charging the battery, the surplus energy is sold to the utility. In this way, the proposed system induces savings related to energy costs without harming the comfort of the user.

Considering that the objective of the proposed work involves the management of surplus energy, it is highlighted in another work, in [53], a surplus energy management system generated by a photovoltaic system using batteries. This case, in [53], differs from what is proposed in the system presented by this work, since it uses energy storage, but it is worth considering the way the authors worked on this system. A Human-Machine Interface (HMI) was developed to allow the user to choose manual or automatic operation modes. In the manual operation mode, the user decides if the system will operate connected to the grid or autonomously. In the automatic mode, the system decides how to connect based on the presence of surplus or not. To develop an interface where the user visualizes the system parameters, such as the operation mode and the existence of surplus, the authors used LabView® 8.5. Another important detail is the ability to generate reports in Excel® to monitor the system operation over time.

## 2.3 Priority of electrical devices

A mechanism used to control energy consumption is the establishment of priority in the operation of loads. Within the context of Home Energy Management (HEM) and considering the Demand Response (DR) based on energy price, in [54], Rastegar et al. present a model using household appliance operation priorities. It is understood that each device does not have the same importance from the consumer's viewpoint. Thus, the categories of controllable devices such as ON/OFF and regulating appliances in the HEM program were considered, as well as Time-of-Use (TOU) and Inclined Blocking Rate (IBR) tariffs. According to these tariffs, it was possible to define the Value of Lost Load (VOLL). For each electrical appliance, this value is related to the load operation priority. Therefore, in addition to the tariff analyzes cited by the authors, the article proved that the insertion of this method of home appliance priorities contributed to a lower energy cost for the consumers without deeply compromising their comfort.

In [55], Shakeri et al. developed a work that connects electrical appliances to smart plugs in communication with the residential HEMS. This one is connected to the utility

company and receives demand response signals. Smart plugs read data such as temperature, current and voltage through sensors and send them to the HEMS controller. Depending on the utility company signal, this controller decreases or alters the operating time of the devices according to energy demand and price. In this context, the HEMS controller uses strategies that establish priorities for device operation. The uncontrollable devices work without interruption while the controllable ones have their operation based on the priority established by the users. In this way, energy consumption can be reduced during peak hours.

In this context, considering that DR represents changes in energy use by the consumer according to the needs of the grid and understanding the ability of the HEMS to participate in the demand response system at the appliance level using smart plugs, in [56], Zhai et al. analyzed the flexibility of appliances, which means the capacity of appliances to participate in DR programs. Thus, the authors used bottom-up method to study the flexibility of the devices and, by means of smart plugs, they identified electrical devices from user behavior and measurement uncertainty. Then, they examined the flexibility of the appliance based on information such as type, power consumption, user behavior, and controllability. By calculating the flexibility of appliances, HEMS can build a priority list and control appliances based on that list. The flexibility calculation favors the application of HEMS without compromising consumer welfare.

## 2.4 Machine Learning in residential energy management

This section will initially discuss concepts and definitions of Machine Learning, and then case studies related to its application in residential energy management.

### 2.4.1 Concept of Machine Learning

As the term suggests, the concept of machine learning is directly associated with the ability of a system to use data or experiments to learn behavioral models and make decisions based on the training obtained, as defined in [57]. In that same cited work, classifications of Machine Learning algorithms are presented. In supervised learning, the output or label of the system is known and the goal is to find a model of the relationship between input and output to predict or classify data. Methods such as Linear Regression and Decision Tree are examples of this type of learning. Unsupervised learning does not use labels to train its data but is based on patterns from the application of measures such as clustering. Semi-supervised learning uses both labeled and unlabeled data for training. Finally, reinforcement learning is characterized by the application of trial and error, which gives rise to a rewarding process based on the feedback provided [57].

### 2.4.2 Machine Learning case studies

In [57], the authors present the use of machine learning in residential energy management. In the context of the electric grid and energy supply and demand, efficient management of residential consumption enables better control of the supply of energy to the grid. Thus, the application of Machine Learning can be adopted to predict household consumption demand, analyze load profiles, and perform load management, enabling reductions in energy consumption and alleviating problems intrinsic to demand-side consumption variations, using methods such as optimum-path forest clustering, k-means, hierarchical clustering and self-organizing map, k-nearest neighbors and gaussian mixture models, factored conditional restricted boltzmann machine, Semi Markov models.

In addition to this theme, in [58] it is understood that Non-Intrusive Load Monitoring (NILM) is a system used to capture the separate consumption of loads. Based on this, the authors used submeters to obtain current and voltage values of the loads separately and machine learning to, based on the data obtained from the readings, predict user behavior. The techniques of the Nearest Neighbor Algorithm and Markov Chain were used. This

machine learning was elaborated in MatLab together with a Graphical User Interface (GUI).

In [59], the authors worked on a strategy for forecasting power generation in a grid-connected photovoltaic system that has a battery bank for storage in a residential building. The intent was to charge the battery one step ahead based on the forecast using the logistic regression method. Based on generation and consumption data, the sliding window approach was used to analyze the regression process, considering the period of 1000 runs. Thus, the authors obtained a significant breakthrough as the system supplies more than half of the annual load demand.

With respect to the application of Linear Regression, in [60], the authors worked on the development of modeling to calculate energy consumption in commercial buildings. In this context, Multiple Linear Regression was applied using 17 predictor attributes for the response Y, which corresponds to the energy consumption in the year, in 7 building shapes. Thus, linear regression was a method used in this case to predict energy consumption.

## 2.5 Used tools

This topic deals with the tools used to execute the stages of this work. Approaches are made to the main characteristics and functions of the resources used to ensure the operation of the proposed system. The Chapter 3 will detail how the system works.

### 2.5.1 ESP32

The ESP32 is a technology with a wide application in the IoT. This device integrates the microcontroller and resources like Wi-Fi and Bluetooth. In [61], there are techniques' specifications about ESP32 that evidence its differential when compared to other modules. Its processing consists of a dual-core system Xtensa LX6, the wireless communication includes the protocol WLAN MAC 802.11 b/g/n/e/i, and the Bluetooth is defined by v4.2 BR/EDR and BLE Bluetooth. Among the various functions of the device, its memory includes 520 KB SRAM, 16 KB SRAM in Real-Time Clock (RTC), 448 KB ROM and its

architecture contains GPIOs that include 12-bit ADCs up to 18 channels, two 8-bit DACs, ten pins that act as touch sensors, three pins that work with UART communication, four pins that can communicate using SPI, two pins with I2C interface, and two pins with I2S interface [62].

In [63], there are also specifications about ESP32, used in the home automation project presented by the authors. This device has a supply voltage of 3.3V and the clock frequency is up to 240MHz. The programs run on it are 32-bit and it is an open-source platform, which favors its extensive applicability in IoT. The authors developed a system where the data read by the LDR light intensity, a Passive Infrared (P.I.R), a DHT22 temperature and humidity sensor, and a MQ-6 sensor were processed by the ESP32. This sends the information to the Firebase database and controls the electrical appliances, based on the data obtained by the sensors. Users also manually control the lights using an Android mobile application, which allows the data to be viewed and monitored. It is worth mentioning that the database stores the data read by the sensors and the data related to the control of the appliances.

### 2.5.2 Message Queuing Telemetry Transport (MQTT) Protocol

MQTT is a Machine-to-Machine (M2M) communication protocol developed by Andy Stanford-Clark and Arlen Nipper at the International Business Machines Corporation (IBM), according to [64]. The authors of this article explain that this protocol uses TCP/IP and has a publish-subscribe structure. This architecture sends messages to a topic from a server, called broker. Clients subscribed to a topic have access to published data. In addition, MQTT is lightweight, fast, and used on devices with low bandwidth and low power consumption. Thus, this protocol can be easily applied in home automation systems.

Furthermore, in [64], the authors mention that MQTT is a fast and secure application protocol that can be used for connection between smart meters, smart appliances, and the power distribution system operator. This ease of communication collaborates with the

ability of power distributors to encourage the use of electrical devices at times of surplus power production.

In [65], the authors declare that MQTT is message centric, which can be flexible but respect the 250 MB size limit. In addition, the overhead of the message is low and this protocol allows the implementation of advanced security measures. MQTT uses the TCP/IP protocol, which maintains a connection guarantee and handles disconnections. In this regard, the authors show an implementation of MQTT in a domestic heating system in the context of Automated Demand Response (ADR). Considering the change of the fuels used for heating systems, especially for electric heating systems, the use of surplus renewable energy is highlighted. In this way, the heating units were connected to the Mosquitto, an MQTT broker, and the connection is maintained through TCP/IP, which allows the utility to know that the device can be controlled. A point to highlight is the fact that this connection can be used to have consumption data and, consequently, demand patterns.

### 2.5.3 Node-RED

In [66], important features of the Node-RED platform are presented. It is a JavaScript-based visual programming tool developed by the IBM. Its server is based on `Node.js` and allows the construction of data processing logic and has an interface where it is possible to visualize sensor data in real-time. Its structure is defined by node, flow and debug panels. Nodes can execute connection functions such as MQTT and TCP. Being an open-source tool, Node-RED is widely used in IoT applications and allows communication between different platforms, including hardware devices, APIs and other services [67].

### 2.5.4 InfluxDB

In [68], one notes the description of InfluxDB as a TSDB. It is written in the Go language and developed by InfluxData, being an open-source platform. InfluxDB can be used in on-premises or cloud applications and has a wide range of IoT and real-time monitoring

applications, as well as providing functions such as alerts and graphical visualizations of data behavior over time. In [69], it is commented that InfluxDB was released in 2013 and is a high-performance non-relational database, enabling high data to write and read speeds. This tool allows integration with numerous libraries and has the ability to optimize storage space.

### 2.5.5 Grafana

According to [70], Grafana® is a web platform that provides interactive data visualization and monitoring created by Grafana Labs. Its visualization tools include graphs and tables, and it also allows the application of alerts and dashboard layouts. Its application can be used in different operating systems, such as Linux, Windows, and MacOS. Companies such as NASA use this tool in certain applications. In [71], Grafana is defined as an open-source application that allows the analysis of information and the display of data, besides allowing integration with other tools such as InfluxDB.

### 2.5.6 Home Assistant

Home Assistant is a tool used in the context of home automation. In [72], Home Assistant is defined as an open-source platform capable of controlling devices and is constantly evolving and expanding its functions. It is an application that can also be used by various operating systems. Among the ways of configuring Home Assistant to monitor and control devices, is the use of Raspberry Pi for its usability and efficiency. In addition, one should highlight Home Assistant features, such as the possibility of creating cards related to the connected equipment in an interactive interface.

Home Assistant is also a tool with great applicability in IoT. In [73], the possibility of monitoring and controlling domestic devices using Personal Digital Assistants (DPA) is demonstrated. In this context, for the implementation of this system, the authors used the ESP8266, whose function is to receive data from the sensors and control the actuators. To send sensor data to the server on the ESP8266 and control relays to turn on and off

certain equipment, an Arduino with built-in Wi-Fi was used, whose communication with the server is done through MQTT. The ESP8266 and the Home Assistant were connected to the MQTT broker and, in this way, the communication between these elements of the system ensured the possibility of receiving information by the Home Assistant and sending commands.

Based on the works and tools mentioned, it is understood the entire scope of application in which the project is inserted, existing technologies in this sense and possibilities for improvements. The next chapters provide further details of the proposed work and its application.

# Chapter 3

# Development

This chapter presents the methodology and the development of the proposed work. Section 3.1 explains in detail the structure of the developed system and specifies the methods used. In addition, Section 3.2 presents theoretical concepts and tools used to apply Linear Regression as a Machine Learning method. Section 3.3 shows the development of monitoring functions and control tests and Section 3.4 presents the study and application of data prediction using Machine Learning.

## 3.1 System architecture

As was observed in Chapter 1, it is understood that it is common to have surplus power generation in distributed productions and that, generally, this excess is destined for batteries or inserted into the distribution network. In the last case, it is known that the consumer receives nothing or an extremely low value for the surplus inserted into the grid. In addition to this fact, one must consider energy losses in the distribution grid, including the energy sent per consumption unit. In addition, in the context of selling surplus, the prosumer must go through certain bureaucratic procedures and have a bi-directional meter in order to be a seller of energy. In the case of batteries, the price of these devices on the market is still high, despite significant reductions in recent years. As a result, many people choose not to subscribe to this equipment.

In this context, the proposal of this project is to direct the surplus energy for use in electrical devices, instead of sending it to the grid, where there are losses, or to batteries. The work presents the development of a system for the measurement and monitoring of surplus power generated and control of power consumption by loads connected to smart plugs, based on the excess power available. In this way, the consumer will be able to visualize data such as excess generation and control devices by means of an interface. This control will not only be ON/OFF but will also occur by managing the operating power of specific loads, such as lamps, heaters, and water heaters, among other loads that admit this type of operation. However, the focus of this work is not concentrated on the ability to control the operating power of a load in a smart plug, but rather on the management of surplus power that will be directed to these devices. Load control is related to surplus power because the power that the system will direct to a load depends on the available power. Thus, the function of the intelligent system is to predict surplus power generation according to the production data of the user, through a Machine Learning analysis method. This process enables the development of automatic decision-making based on excess generation and consumption history.

For a better understanding of how the proposed system works, Figure 3.1 presents a schematic of the components used for the development of the work.

As presented in Figure 3.1, the ESP32 microcontroller is used to capture the surplus energy and consumption data. In this work, more focus is given to the surplus energy analysis. However, user consumption data is important for implementing decision-making in the system. The ESP32 takes voltage, current and power readings, and sends the excess power data via WiFi to a topic on the MQTT broker. Node-RED is used to process the information. Node-RED accesses the MQTT topic and sends the power values to the database in InfluxDB. These values can be visualized via the Grafana monitoring platform and are used by Machine Learning tools for data analysis and predictive development. It can be seen from the schematic that the Node-RED, InfluxDB, and Grafana applications are used in integration with Home Assistant. This platform also allows data visualization through an interface. Regarding control, the user can remotely control a load via the Home

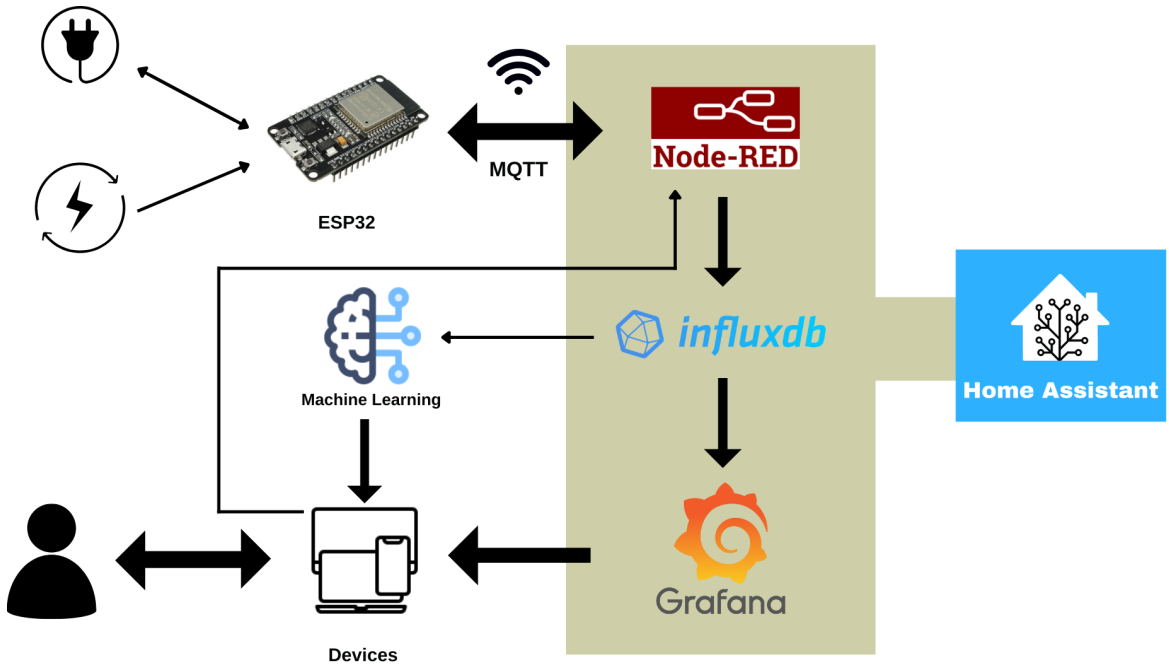


Figure 3.1: Schematic of the proposed system.

Assistant interface. The command is processed by the Node-RED, stored in InfluxDB, and sent to the microcontroller that belongs to the smart plug. In addition, the control can also be done automatically, with decision-making based on Machine Learning [74].

## 3.2 Machine Learning

In addition to the definition given in Chapter 2, Machine Learning is based on computational algorithms that allow automatic decision-making from data [75]. Among the known methods of Machine Learning, Linear Regression, a method within the supervised learning classification, was applied in this work. As presented in [76], the Linear Regression is a relation given by Equation 3.1.

$$y = a_0 x + a_1 + e \tag{3.1}$$

where $y$ represents the dependent variable and the value to be predicted, $x$ is the

independent variable, $a_0$ is the angular coefficient, $a_1$ is the linear coefficient, and $e$ represents the random error. Also, in [76], the authors apply the Python language to realize data prediction by Linear Regression. They mention libraries that are important in this regard to guarantee the application of this method in Python, such as the Pandas and Scikit-Learn libraries. This approach will also be used for the proposed work. By applying Linear Regression with excess generation data, some parameters will be observed, such as accuracy, which measures the accuracy of the model in predicting the correct result, and mean absolute error, which is the average error between the predicted and the actual values. The Mean Absolute Error (MAE) is given by Equation 3.2 and was applied as the quotient of the sum of the error modules and the number of samples that would be predicted.

$$MAE = \frac{\sum |y_{real} - y_{predicted}|}{number\ of\ samples} \tag{3.2}$$

where $y_{real}$ is the real values read from Y and $y_{predicted}$ is the predicted values. Another parameter is median absolute error, which represents the median between predicted and real values. Equation 3.3 presents this relation.

$$MdAE = median(|y_{real_1} - y_{predicted_1}|, ..., |y_{real_n} - y_{predicted_n}|) \tag{3.3}$$

where $y_{real_1}$ represents the first real value of Y among the samples; $y_{predicted_1}$ the first predicted value of Y among the samples; $y_{real_n}$ the last real value of Y among the samples and $y_{predicted_n}$ the last predicted value of Y.

In addition, another parameter is explained variance score, which measures the ability of the model to explain the variation in the data and the closer the model is to 1 or 100% the better it is. Equation 3.4 shows how to calculate this parameter.

$$explained\_variance\_score = 1 - \frac{Var(y_{real} - y_{predicted})}{Var(y_{real})} \tag{3.4}$$

where $Var$ represents the variance of the considered values.

Finally, there is error variance, which represents the measure of the dispersion of the forecast errors relative to the mean of the errors. Considering that the error is given by the difference between the real value ($y_{real}$) and the predicted value ($y_{predicted}$), Equation 3.5 shows how to calculate the error variance.

$$error\_variance = \frac{(error - \frac{\sum error}{number\ of\ samples})^2}{number\ of\ samples} \tag{3.5}$$

The development environment that will be used for Python programming is Visual Studio Code.

In this way, it is possible to understand in this section the function of each component within the system structure of this work and the methods that will be applied so that the development of the work reaches the proposed objective.

## 3.3 Monitoring and control system configuration

To begin the analysis of the surplus energy generated, a measurement simulation was created to understand the flow of electricity in a residence. Before explaining this implementation, it is important to highlight the profile presented by the electric current in both situations: when surplus energy is generated and the generation system is connected to the grid, the current follows the same direction as the grid voltage and when there is the consumption of grid energy, the current follows the opposite direction of the voltage. With this analysis, it will be possible to investigate if there is surplus power or not. In this section, initial tests will be presented with the tools used for data storage and visualization, as well as device control. It will also be presented the development of a prototype responsible for measuring and processing the surplus power data generated and sending it to the monitoring system.

### 3.3.1 Initial control tests

Before applying the monitoring of readings related to surplus power, device control tests were also performed on the Home Assistant platform. Basically, the test consisted of turning a LED on and off. In more detail, a button card was created on the Home Assistant control panel called "ON/OFF Led". When clicks were made to turn the led on or off, this command was processed by Node-RED, from a Home Assistant integration node named `events: state`, whose parameters relate to the "ON/OFF Led" entity created in Home Assistant. To this node, a function node is connected which aims to set the value "0" for led off and "1" for LED on. Another node that characterizes the MQTT connection receives these values and sends them to the topic "topico_led" of the broker and to the database. The construction of the Node-RED program is shown in Figure 3.2. Via Wi-Fi, the information from the broker was received by the ESP32 which, based on the received data "0" or "1", turned the LED on or off.

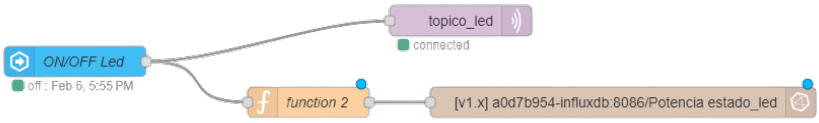


Figure 3.2: Command processing by Node-RED for LED turning on and off.

### 3.3.2 Surplus power monitoring

To understand the application of the data storage and monitoring tools, tests were made on voltage and current readings. Using the Arduino Integrated Development Environment (IDE) as a development environment and the C programming language, the data was obtained by the ESP32 microcontroller. Then, the information was sent over WiFi to an MQTT broker and processed by Node-RED, which send to the InfluxDB database. After, the data was made available for visualization in Grafana.

After initial tests with Node-RED, InfluxDB, and Grafana, studies and usage tests were made of the Home Assistant platform. This tool was installed on a virtual machine and, considering its ability to integrate with other applications to expand its functionality,

Node-RED, InfluxDB, and Grafana were installed on the Home Assistant. Thus, the data processed by Node-RED and stored in InfluxDB can be presented in the interface of Home Assistant.

To perform surplus power monitoring, prototype hardware was built to perform electrical voltage and current measurement and, based on the current analysis, define whether or not there is surplus power and its value. Initially, however, surplus power was simulated by means of a variable resistor connected to the electrical grid for testing purposes. Based on this, three tests were performed to perform this measurement: simulation of surplus power with a variable resistor, grid power reading, and photovoltaic generation power reading. The last two were performed to have a comparison of the current analysis when energy is absorbed or injected into the grid.

### Simulation of surplus power with a variable resistor

To read the voltage, the prototype hardware built used a 68 Ω power resistor connected to a 12V transformer plugged into the 230V power grid. To reduce the 12V output voltage of the transformer and allow the voltage to be read at the input port of the microcontroller, resistors were used as voltage dividers and capacitors for signal coupling. Figure 3.3 shows the RMS grid voltage read by the oscilloscope and parameters such as peak voltage and frequency.

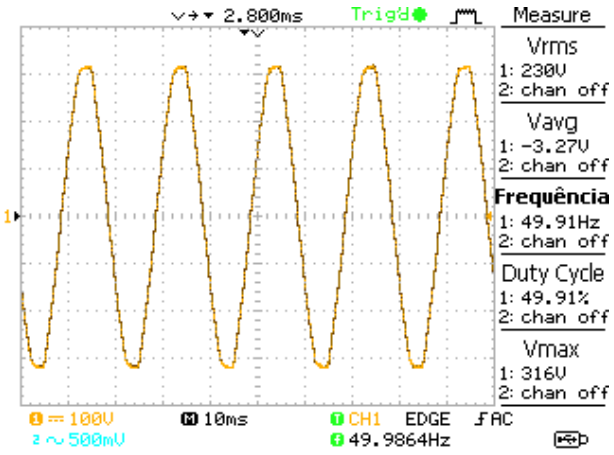


Figure 3.3: Grid voltage reading.

Based on the voltage divider circuit and with signal coupling, the voltage signal that is fed into the input port of the microcontroller is shown in Figure 3.4, with a reading taken on the oscilloscope.

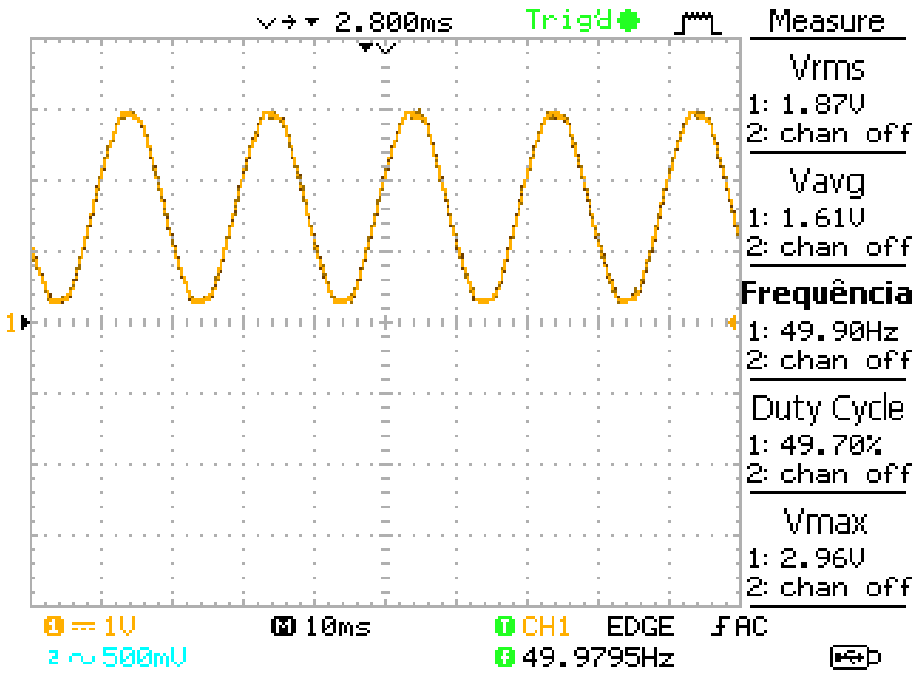


Figure 3.4: Reading the input voltage on the microcontroller.

The voltage values are taken in 1000 sample intervals by the microcontroller into a *for* function in programming, as can be seen in the appendix B, and also corrected in the program to be analyzed on the grid voltage scale. With the 1000 samples it is possible to calculate the RMS voltage of the grid and check it against the values read in the oscilloscope.

For the electrical current reading, the ACS712 30A sensor was connected in series to the source, which in this case is the 230V/12V transformer connected to the grid. The current sensor was also connected to a variable resistance of 42 Ω and nominal current 5A. This procedure was performed to simulate the variation of surplus power in the measurement system and evaluate the precision of the measurement. Figure 3.5 shows the construction of the surplus power simulation system.

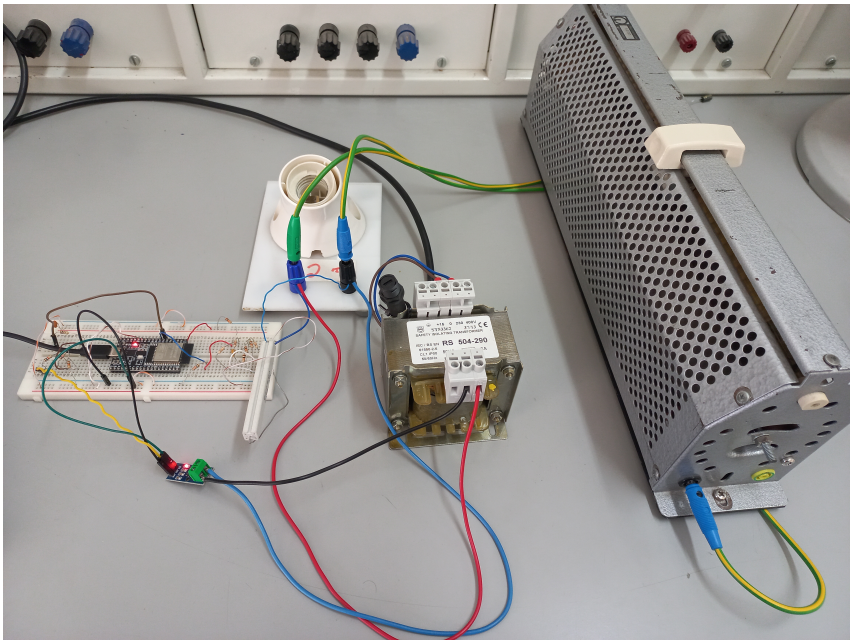

Figure 3.5: Construction of the prototype for simulating excess power using variable resistance.

In obtaining the electrical current and voltage values, intervals of 1000 samples were set in the code. It should be noted that measuring equipment, such as an oscilloscope, was used to check the certainty of the measurements. These voltage and current values are sent to the topics "topico_tensao" and "topico_corrente" respectively in the MQTT broker and these values are stored in the InfluxDB database. Figure 3.6 shows the Node-RED structuring for the execution of these functions.

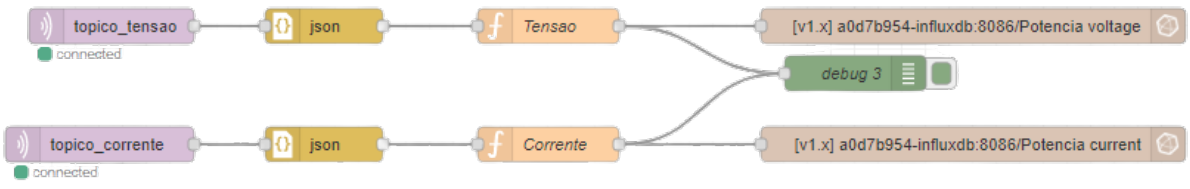


Figure 3.6: Building processing of voltage and current data and InfluxDB storage by Node-RED.

Thus, with the readings taken, 1000 power values were determined, and then it was possible to define the RMS power, which simulates surplus power value. This power

data is sent over Wi-Fi to the “topico_potencia_excedente” topic in the MQTT broker. Node-RED accesses this topic and processes the information, sending it to the InfluxDB database in Home Assistant, as shown in Figure 3.7. In this way, it is possible to visualize the excess power data in the Home Assistant interface itself and also use resources such as Grafana.

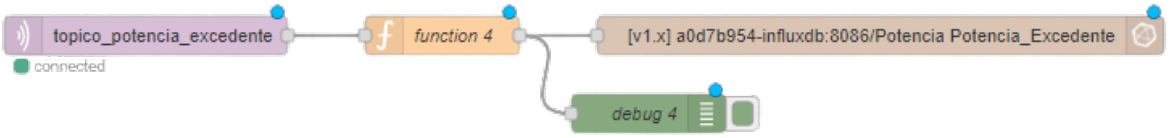


Figure 3.7: Building processing of power data and InfluxDB storage by Node-RED.

To define the surplus power value to be directed to a load, a relation between surplus and load demand was constructed, as an initial test. For this, the Home Assistant was used, whose interface the user uses to inform the power value of the load to which this power must be directed. Figure 3.8 shows an initial example of the interface where the user can enter the power consumed by the load into the Home Assistant. In the same interface, there is also the example of the led control mentioned previously.

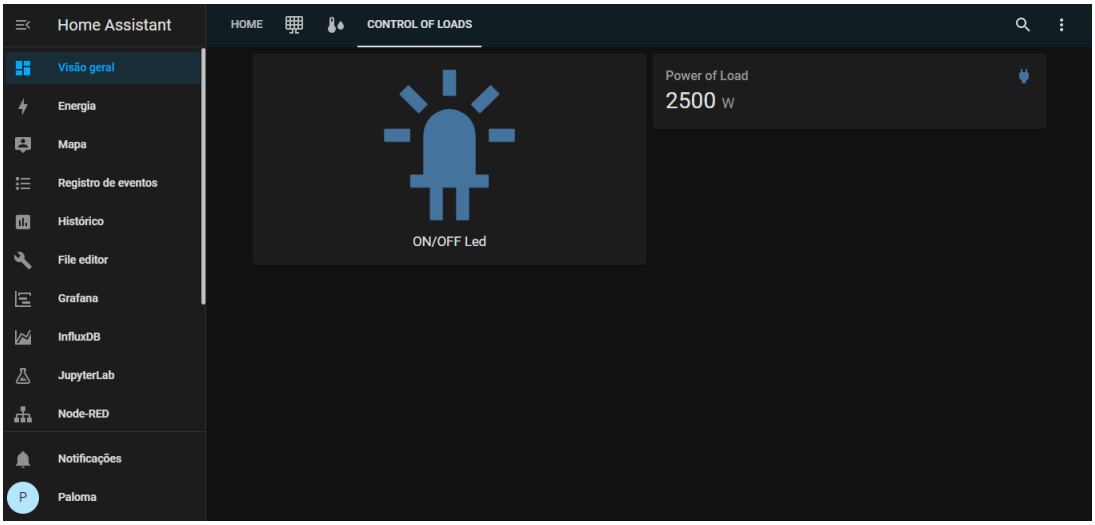


Figure 3.8: Simplified example of an interface in Home Assistant that allows the user to enter the load power.

Naturally, this type of manual input by the user is subject to error. However, this example has been taken only as an initial consideration for a better understanding of how the system works. This informed power value is processed by Node-RED and sent to the topic "topico_potencia_carga" of the MQTT broker, as shown in Figure 3.9.

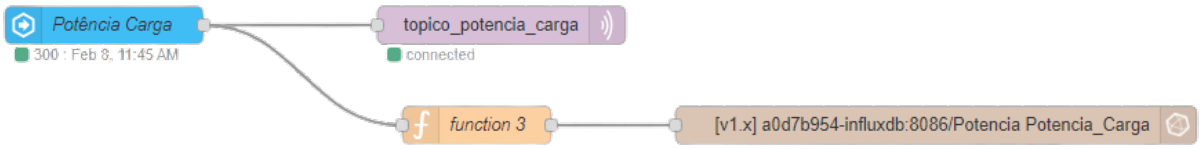


Figure 3.9: Processing of the load power data by Node-RED.

Through Wi-Fi, this value is received by the ESP32. Thus, with the excess RMS power value obtained with the excess generation simulation and the power value demanded by the load, informed by the user, the relation between both values in percentage is calculated. It is a simple relation given by Equation 3.6.

$$p = \frac{P_{exc}}{P_{load}} 100 \tag{3.6}$$

where $p$ represents the percentage ratio between the powers, $P_{exc}$ represents the excess power and $P_{load}$ equals the power of the load.

The value of this percentage is sent to the smart plug that controls the load and defines the percentage of its operating power. In this sense, control tests were carried out that considered loads whose operating power variation is allowed, such as lamps and heaters.

### Grid energy consumption reading

After simulating the surplus power generation, measurements and graphical representations were made of the voltage and current data obtained from the grid-connected hardware prototype. In this case, the system load is, in fact, the 42 Ω variable resistor used in the previous application. The voltage and current readings were taken in the same way and with the same devices as the measurements taken previously. The purpose of this analysis is to verify the current behavior compared to the grid voltage and to ascertain the

lack of surplus since it is the grid that supplies power to the system. The results obtained are presented in the next section and provide the basis for a comparative analysis of the occurrence of excess generation.

The values of the voltage and current samples are used to calculate the power samples in the code. The Appendix B presents the code used for the measurements and communication with the MQTT broker. It is noted in this appendix that the power sample values are negative in this case. However, the calculated RMS power results in a positive value and considering the approach mentioned in the previous topic about the percentage calculation that relates surplus and demanded power and understanding that this surplus power value in this relation will always be positive, treatment was done in code so that the negative power could represent in fact the lack of surplus. Based on the power sample values, counter variables were created. If the power sample was a negative value, the $P_{neg}$ counter was incremented (according to Appendix B). If it was a positive value, the $P_{pos}$ counter was incremented. Thus, if $P_{neg}$ was higher than $P_{pos}$, the power was, in fact, negative and thus there was no surplus and the value of the percentage $p$ would be 0. Otherwise, the power was positive and there was surplus, then the value of $p$ is higher than 0. This relationship works properly, especially in the considered applications with resistive loads. A refinement of this analysis can be applied to the evolution of this prototype. It should be considered that the excess power value, whether it is null or not, is sent to the MQTT broker and processed by the Node-RED based on Figure 3.7.

**Photovoltaic generation power**

The power reading generated in one of the inverters of the photovoltaic system of the *Escola Superior de Tecnologia e Gestão (ESTiG)* at the *Instituto Politécnico de Bragança (IPB)* was developed. About 9 photovoltaic modules of nominal power 220W are connected to the inverter in question, which is connected to the electrical grid. In this sense, the LTS-25-NP current transducer was connected to the inverter. Thus, the current reading is done by this device and the voltage reading remained in the same structure as the topic of surplus power simulation with variable resistance. Figure 3.10 presents the

adaptation of the prototype mentioned above for the proposed system, considering the connection of the LTS-25-NP in series with the inverter and the electric grid.

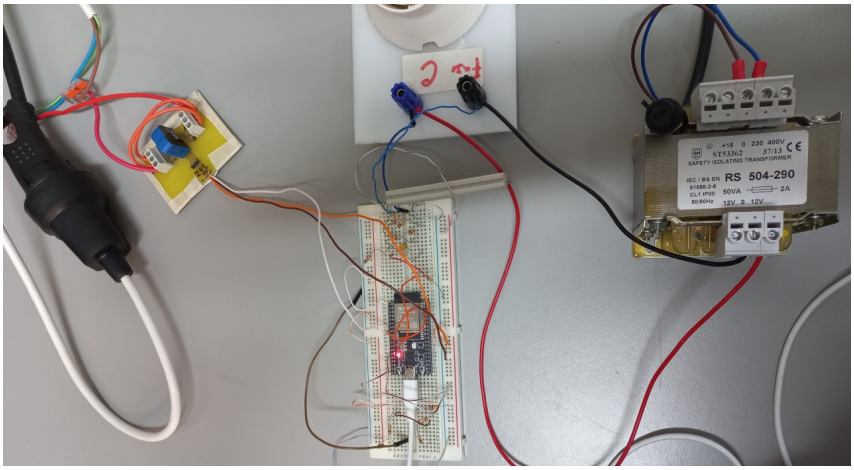


Figure 3.10: Prototype for current and voltage measurement connected to a photovoltaic inverter.

The voltage, current and power values are processed by the Node-RED, as shown in Figures 3.6 and 3.7, and sent to the database. With the information in InfluxDB, it is possible to access and view it through the monitoring platforms.

With regard to the analysis of the power samples for the verification of surplus power, that is, to verify if the energy is being inserted or imported from the network, it was used in this case. The results obtained will be presented in the next chapter. The same verification applied to grid power data and presented in Appendix B was used. It should be noted that in [77], the LTS 25-NP transducer reads a nominal RMS current of 25A when it has 1 primary turn in its structure. As can be seen in Figure 3.10, the sensor under consideration has 3 primary turns, which gives it the ability to read RMS current of 8A. It is for this reason that the programming placed in the Appendix B for the signal reading of the LTS 25-NP transducer considers the division by 3, when the current reading is done by this device.

## 3.4 Machine Learning: Application of linear regression

Considering surplus power data generated in distributed energy production, an analysis was made with data from one year of generation. As this project was developed throughout this time, including the use of prototype hardware for measuring voltage, current, and power, it was not possible to capture one year of data. Thus, to perform the implementation of the data prediction by the Linear Regression method, an existing dataset was used and made available for use.

### 3.4.1 Pre-analysis of the dataset

Data analysis was done using the readings from the `Package Household Data` database made available in [78]. The *household_data.xlsx* file made available in Microsoft Excel was used. It is important to mention that the worksheets contained readings of energy consumption by specific loads and photovoltaic generation in kWh of various consumption units, from industries to households in southern Germany overtime at 15 and 60 minute intervals. Focusing on the data with 15 minute reading intervals, it is observed that there is information from December 11, 2014, at 5:45 pm to May 1, 2019, at 10 pm. For a more detailed study and in order to facilitate the understanding of the application of the forecasting method, it was chosen the consumption unit defined in the spreadsheet as *DE_KN_residential6* and data between the date of May 14, 2016, at 00:00 am and May 14, 2017, at 11:45 pm with 15-minute intervals between readings. These data were organized in just one worksheet in an Excel worksheet. Although this data is organized in Excel, the purpose of using storage platforms such as InfluxDB and monitoring such as Grafana refers to the future use of surplus energy data read by the developed energy metering prototype.

Initially, it is given that *utc_timestamp* is the time period in Coordinated Universal Time. Also, it is observed that, in the *DE_KN_residential6* of the dataset, there is a

study of consumption of 4 specific loads: circulation pump, dishwasher, freezer and washing machine. In this way, *DE_KN_residential6_circulation_pump* represents a numerical quantity of the float type that shows the energy consumption of a circulation pump in kWh in a residential building in an urban area; *DE_KN_residential6_dishwasher* is a numeric quantity of the float type representing the energy consumption in kWh of a dishwasher in a residential building located in an urban area; *DE_KN_residential6_freezer* is a numeric parameter representing the energy consumption in kWh of a freezer in a residential building located in an urban area; *DE_KN_residential6_washing_machine* is a numerical quantity of the float type representing the energy consumption in kWh of a washing machine in a residential building in an urban area; *DE_KN_residential6_grid_export* is a numerical parameter of the float type representing the energy in kWh exported to the grid in an urban area by a residential building; *DE_KN_residential6_grid_import* is a float type numerical parameter that represents the energy in kWh imported from the grid in urban area by a residential building; *DE_KN_residential6_pv* is the total photovoltaic energy generated on a residential building in kWh in an urban area. The energy values in all these parameters shown are cumulative. In this way, additional columns were created in the worksheet to show the energy values at that given time, which meant subtracting the value of the cumulative sample in question from the previous one. In this way, the values of energy generated, imported and exported from the network and consumed by the loads in each specific time, in intervals of 15 minutes, were obtained. Furthermore, after obtaining these new data, it was necessary to define the surplus power data. For this, the values obtained from the consumption of each load were subtracted from the value of photovoltaic generation. Note that the relation between generation and consumption is very discrepant, as the consumption of loads is reduced to the production of available energy. Figure 3.11 demonstrates this relation.

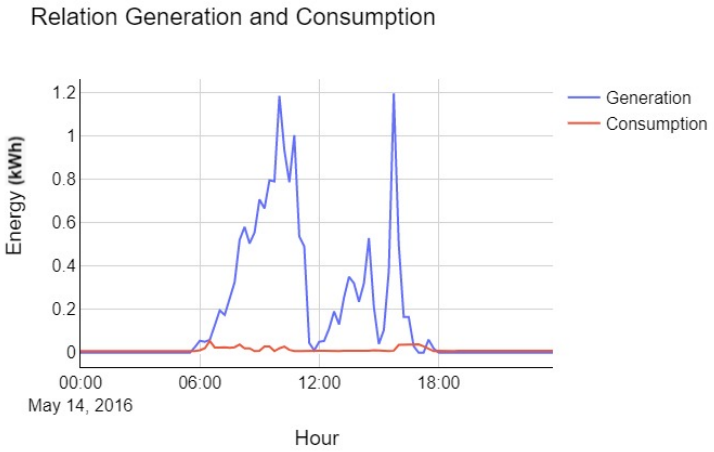


Figure 3.11: Relation between photovoltaic generation and energy consumption by circulation pump, dishwasher, freezer and washing machine in the residence under analysis.

Figure 3.11 shows a representative graph of generation and total consumption of the loads considered in the residence on May 14, 2016. Considering the discrepancy between generation and consumption values, it can be concluded that the relationship between surplus energy and consumption will represent a significant difference, although there are periods of the day when there is more consumption than surplus. Figure 3.12 shows this relation.

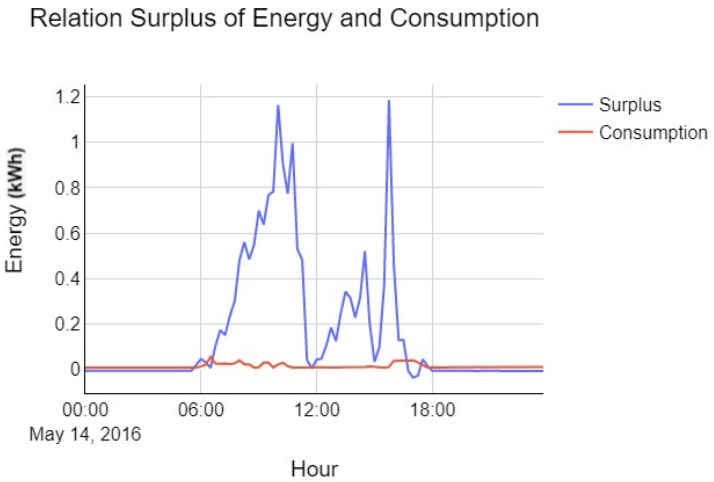


Figure 3.12: Relation between surplus energy and energy consumption by circulation pump, dishwasher, freezer and washing machine in the residence under analysis.

It is possible to observe in Figure 3.12 that before 6 pm on May 14, 2016, there are periods in which the total consumption of the loads considered is greater than the surplus energy. Although this is an atypical case, in which most of the time there is surplus production, the data analysis considered this dataset available for study in order to understand the application of the forecasting method.

### 3.4.2 Application of Linear Regression

Using the Python programming language and the Visual Studio Code platform as a programming environment, tools such as the Pandas library were applied to the dataset to create a dataframe from which data analysis would be performed. Appendix D presents the programming code in Python with all the data analysis process done in this work. After creating the dataframe, the information was inserted into the database by converting sample data into a line protocol, a format accepted by InfluxDB. Thus, when converting the data to line protocol, a text file was created with the data in that format, the connection was established with InfluxDB and the information was inserted into the database.

After the information was inserted into the database, the data analysis began. The collection of samples that will be used to create the Linear Regression model and the data that will be predicted is consulted to perform, after the model implementation, the comparison of the actual values with the predicted values. Regarding the linear regression application in Python, it was performed using the libraries `StatsModels` and `Scikit-Learn`. As mentioned, linear regression has a dependent variable $Y$, which represents the prediction objective, and an explanatory variables $X$, which corresponds to the predictor attributes.

For the definition of the values of $X$, in [79], an approach focused on the inclusion of new features is noted. The authors worked on weather forecasting, with measurements of maximum and minimum temperatures, using data from the last two days. Thus, in [80], a similar approach is applied in Python. In this case, the author has increased the

resources and the number of previous days for the application of this analysis. In the case of this work in which the sampling interval is 15 minutes and not days, new resources will be included that represent previous measurements considering 15 minute intervals. Thus, observing the data visually organized in an Excel worksheet, for example, the values of $X$ represent the values of $Y$ shifted cells down, and the number of shifted cells refers to the number of new features included. Appendix C presents the data for May 14, 2016, taken from the dataset under analysis for this paper, and shows the inclusion of 6 new resources representing the values of $X$. Looking at the example table in Appendix C, it is necessary to understand what the new features $X1$ to $X6$ mean. As mentioned, each column from $X1$ to $X6$ represents previous measurements of surplus energy. The more offset from the $Y$ values the column is, the more past measurements it represents.

Using the *StatsModel* library, it was possible to acquire parameters such as the coefficients of the proposed statistical model. Then, the approach adopted for data analysis used the `Scikit-Learn` library and was based on separating the values of $X$ and $Y$ in the test and training data. Thus, 80 % of the $X$ and $Y$ values were defined as training data and 20 % as test data. This separation was done using the `train_test_split` function from the `Scikit-Learn` library in Python, as shown in the appendix D. With this classification, the training data is passed as parameters to the Linear Regression model. Based on the training data, the `predict` function can use the test data to make predictions based on the model created. It is possible to highlight that there was no analysis of time variables, for example, only of surplus energy.

With the model defined, it is possible to acquire the coefficients of the Linear Regression function. The angular coefficient represents the slope of the line that defines the function, and the linear coefficient shows the point where the line intersects the Y axis. Thus, once the Linear Regression model was defined and its coefficients were known, the function of the line with the model coefficients, according to Equation 3.4, without random error and, from a *for* function, the real data of Y considering previous measurements are inserted in $X$. These data were previously queried in InfluxDB and inserted into a separate dataframe. With the coefficients and $X$ values, it is possible to find the $Y$ values,

which correspond to the predicted values. It should also be taken into account that the system should not predict negative values. Therefore, treatment was implemented when the result of $Y$ is negative. That way, every time $Y$ is less than zero, it becomes zero. In addition, other important parameters for data analysis were studied, such as accuracy, to measure the prediction efficiency of the model; MAE, to measure the mean of the absolute error values; Median Absolute Error (MdAE), to measure the median of the absolute error values; error variance, to measure the error variation; and explained variance score, to measure how much the model explains the variation in data. These steps can be seen in Appendix C.

The analyses for surplus power prediction were done considering different amounts of samples. The following topics will show the structuring of the study done in this work.

### 3.4.3 Forecast with hours

Considering that the first day of the dataset is May 14th, 2016, a study was done by previous definition with the 1st, 7th, 14th and 21st days from the 14th in which the predictive ability of a model created with 1 hour, 6 hour, and 12 hour data were analyzed. Thus, for May 14th, for example, which corresponds to the 1st day, a model was created with the initial 1 hour of the day to predict data for the rest of that same day; another model with the initial 6 hours of the day to predict the rest of that day and the following day; a model with the initial 12 hours of the day to predict the rest of the day and the following day. This same process was redone for May 21st, May 28th, and June 4th, which correspond to the 1st, 7th, 14th, and 21st days of the first month under consideration, respectively.

In addition, one should also consider that the models were created in all these time cases for different numbers of new features of $X$, that is, models were created only with $X1$, which represents the previous measurement; another with $X1$ and $X2$; another with $X1$, $X2$ and $X3$; and so on, up to $X1$ to $X6$ in order to test how many new features inserted into the model improve the prediction performance. After this study, the interval

of 12 hours has been defined for the forecast, and the model defined by this range of hours was applied to predict the initial 31 days of the dataset year. The results obtained from this analysis are considered in Chapter 4.

### 3.4.4 Forecasts from one day to the next

After the investigation into the efficiency of forecasting of surplus energy from hours, experiments were performed with a full day of data forecasting the next day. These tests considered the first day of the first month of the dataset, i.e., May 14th, to predict the later, May 15th; May 29th to predict May 30th; June 3rd to predict June 4th, and June 13th to predict June 14th. It is notable that the choice of days for prediction was made within the first month of the year under consideration in the dataset, just to start checking the predictive effectiveness in this situation. The implementation of this study was performed along the same lines as the analysis of hours, considering tests with 1 to 6 features for $X$ and calculating parameters such as accuracy, mean absolute error, median absolute error, error variance and explained variance score. The results obtained are presented in Chapter 4.

### 3.4.5 Using 5 days for forecast

An experiment was carried out, by previous definition, using May 14th-18th to create models based on 1 to 6 features for $X$ in order to predict the 6th day, May 19th. The volume of data to train the model is relevantly higher than the other analyses made up to this point of work. This means that in a real context of the efficiency of a model like this, the forecasting system would require 5 days of data capture to perform the forecast. Factors like this explain the range of experiments done in this work, from hours to data days. In the context of this experiment, Figure 3.13 presents the relation of $Y$ and $X$ values, considering 6 features, used for the creation of models. It should be considered that *mean_surplus* corresponds to the real values of Y; X1 corresponds to the real values read from Y considering 15 minutes before; X2 corresponds to the real values read from Y

considering 30 minutes before; X3 corresponds to the real values read from Y considering 45 minutes before; X4 corresponds to the real values read from Y considering 60 minutes before; X5 corresponds to the real values read from Y considering 75 minutes before and X6 corresponds to the real values read from Y considering 90 minutes before.

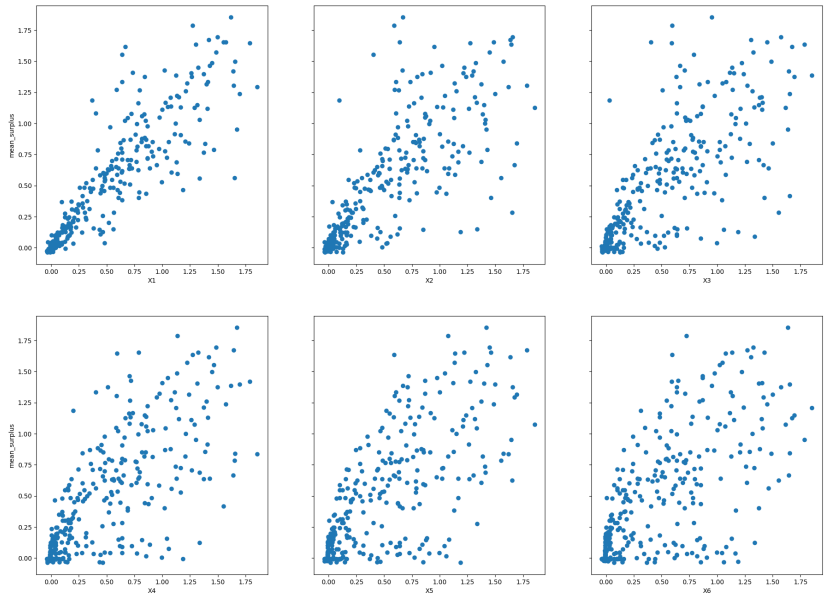


Figure 3.13: Relation between Y and X values used to create forecast models with May 14th-18th.

It is possible to observe that the relationship between $Y$ and $X$ is more linear when the model uses data that represents a previous measurement or 1 feature, that is, $X1$, and becomes less linear as it approaches feature 6 or $X6$. Chapter 4 presents the results obtained from this study.

### 3.4.6 Using 10 days for forecast

In this study, 10 days were used to create the models with different features for $X$, from 1 to 6. Initially, based on the surplus generation data from May 14 to 23, 2016, the 11th day, May 24, was predicted with models containing 1 to 6 features. Then, one of the models with a specific number of features was adopted to predict the same day 24 in 11 months.

With regard to the analysis of the association between the values of $Y$ and $X$ for the creation of the models, Figure 3.14 shows this relation.

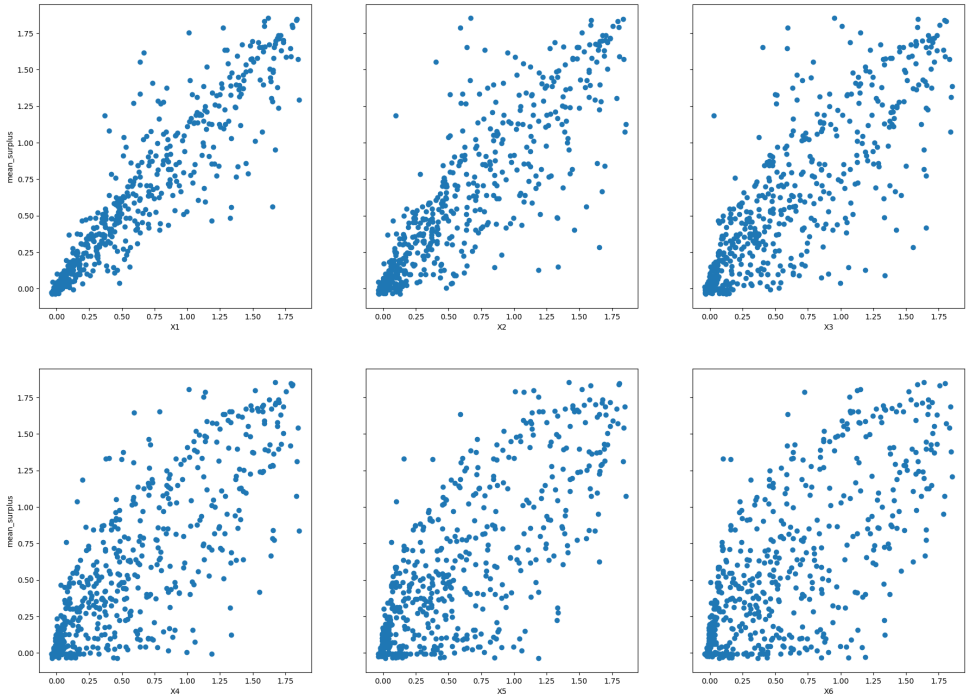


Figure 3.14: Relation between Y and X values used to create forecast models with May 14th-23th.

It is reasonable to conclude a higher linear approximation between $Y$ and $X1$ than with the other features. Considering that these data will be used to create the 10-day forecast models, Chapter 4 will demonstrate the results and analyses obtained.

### 3.4.7 Using 30 days for forecast

A final analysis was performed using 30 days, between the dates of May 14 to June 13, 2016, to predict the 31st day, i.e. June 14, 2016, with models using 1 to 6 features for $X$. Again, it is possible to note a large volume of data to perform a one-day forecast. With respect to the relationship between $Y$ and $X$, Figure 3.15 presents the graphical representation between $Y$ and each feature of $X$ separately.

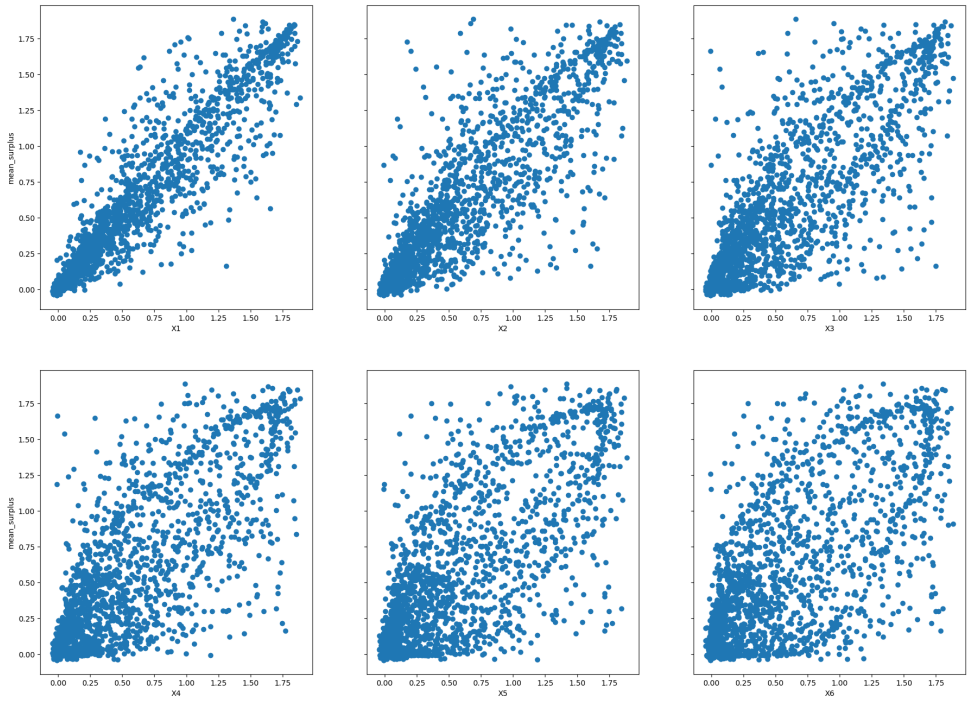


Figure 3.15: Relation between Y and X values used to create forecast models with May 14th - June 13th.

It can be seen, again, that the relation between $Y$ and $X$ is closer for feature 1 or $X1$ and moves away as the feature represents more previous measurements, as is the case for $X6$. In general, however, the relation between $Y$ and $X$ exhibits behavior close to a linear definition.

This chapter presented the development of the proposed project, which included knowledge and application of the data processing, storage, and monitoring tools; construction of a power measurement prototype; device control experiments based on available power, and the analyses that were performed to predict excess power generation through Linear Regression.

# Chapter 4

# Results

This chapter shows the results achieved with the tests made and the satisfactory answers obtained in the procedures based on the objectives expected for this work. Section 4.1 presents the results obtained from the reading of voltage, current, and power by the hardware prototype developed as well as the processing, storage, and viewing of this data and Section 4.2 shows the results obtained with data prediction experiments through Linear Regression and the related analyses.

## 4.1 Surplus power monitoring

The results obtained from the power monitoring tests will be presented, from the simulation with the variable resistor to the application of the measurement prototype on a photovoltaic inverter.

#### Simulation of surplus power with a variable resistor

Considering the simulation of excess power using a variable resistor, Figure 4.1 shows RMS voltage readings stored in InfluxDB. It can be seen that the values do indeed correspond to the 230V RMS value from the power grid.

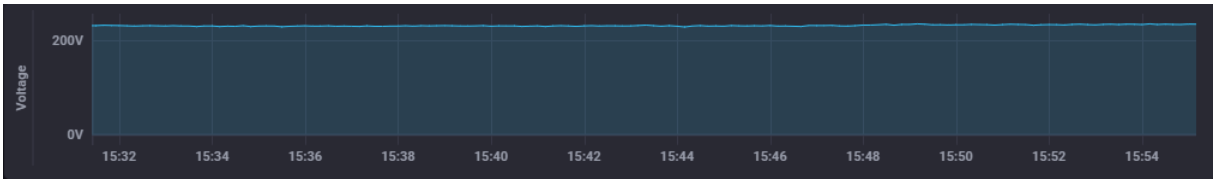


Figure 4.1: RMS voltage data measured by prototype and stored in InfluxDB.

Concerning electric current, the readings taken correspond to the RMS values of the current flowing in the variable resistor. Figure 4.2 shows current data stored in InfluxDB.

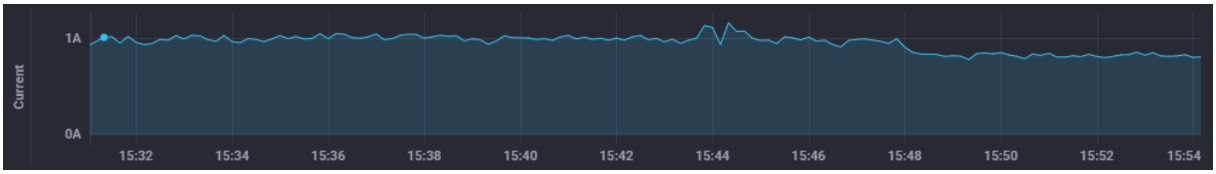


Figure 4.2: RMS current data measured by prototype and stored in InfluxDB.

It can be seen that the current values are around 1A. However, starting at time 3:48 pm the current reduces because the value of the variable resistor was increased manually. Therefore, considering this variation in current, the power value also reduces, since the voltage is maintained. Thus, Figure 4.3 displays the simulated excess power values stored in InfluxDB.


Figure 4.3: Simulated surplus power data measured by prototype and stored in InfluxDB.

It can be seen that the power values are significantly above 200W. Again, starting at time 3:48 pm the values reduce because the variable resistance was manually modified, which reduced the electric current. Observing the power behavior by Grafana it is possible to analyze again these same results. Figure 4.4 presents this visualization by Grafana.

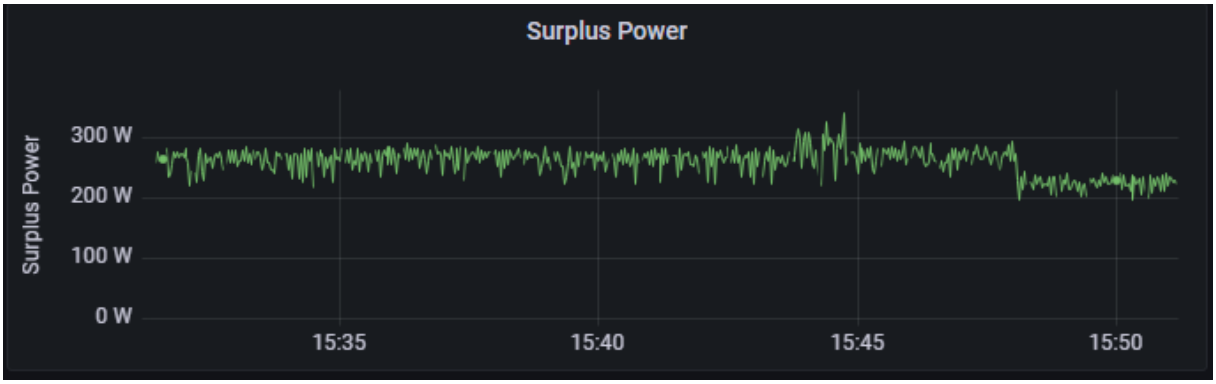


Figure 4.4: Simulated surplus power data measured by prototype and viewed at Grafana.

As mentioned previously, in the Home Assistant screen it is possible to visualize the information related to power. Figure 4.5 exemplifies one of the visualization models from the on-screen display of the excess power value.

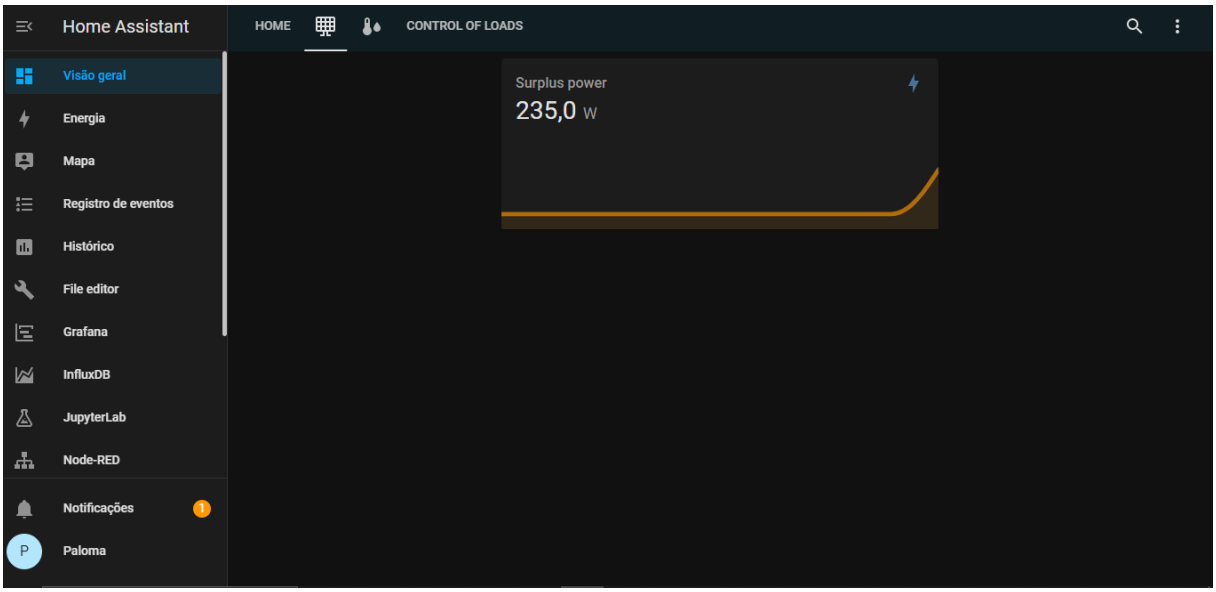


Figure 4.5: Simulated surplus power data measured by prototype and viewed at Home Assistant.

### Grid power reading

As for the voltage and current measurements on a load that consumes energy from the grid, i.e., in a situation where there is no surplus energy, it can be seen that the active power measured is negative, because there is the consumption of electricity from the grid.

Figure 4.6 graphically represents the result obtained with 200 voltage, current, and power samples within a reading of 1000 samples taken.

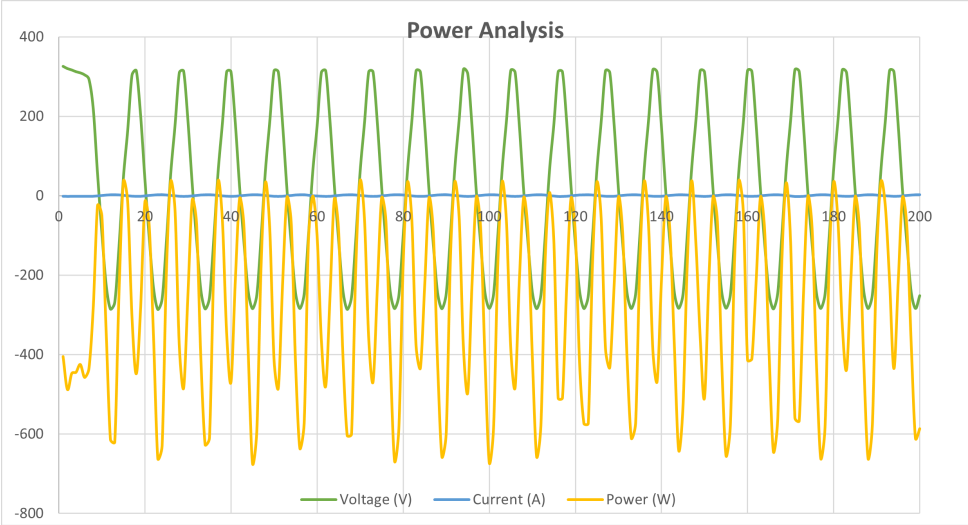


Figure 4.6: Graphical representation of the voltage, current, and power values at a load connected to the grid.

Considering the value of the electric current significantly lower than the voltage and power values, the current signal was graphically represented to facilitate the visualization of the values, as presented in Figure 4.7.

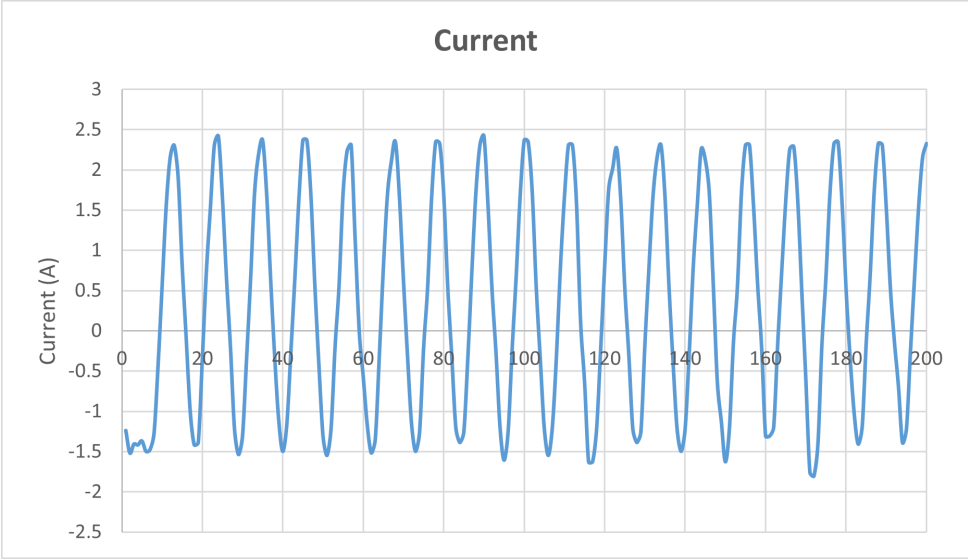


Figure 4.7: Graphical representation of the electric current in the load supplied by the electrical network.

It can be seen that the peak voltage values in Figure 4.6 approach 320V in the positive semicycle and -280V in the negative semicycle. The peak current values in the variable resistor approach 2.5 A in the positive semicycle and -1.8 A in the negative semicycle, as seen in the graph in Figure 4.7. The power values are mostly negative with negative peak values near -680W.

### Photovoltaic generation power

In the context of measuring the voltage and electric current of the prototype connected to the photovoltaic inverter, Figure 4.8 presents measurement results of these variables obtained with 200 samples, within a reading of 1000 samples. It is possible to observe that in a situation of surplus power insertion in the electrical grid, the measured active power is positive.

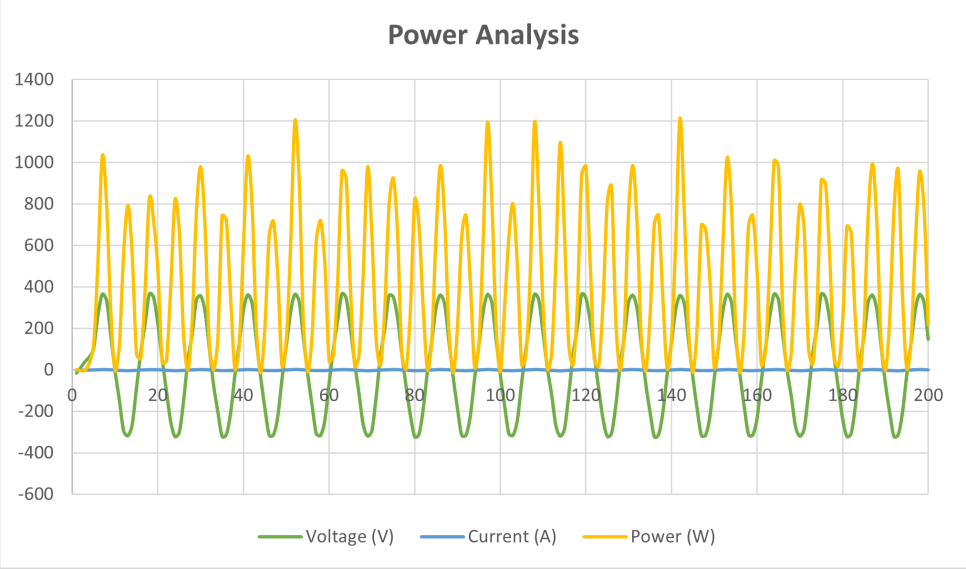


Figure 4.8: Graphical representation of the voltage, current, and power values in a photovoltaic inverter.

Again, considering the current values significantly lower than voltage and power, Figure 4.9 shows in more detail the resulting current signal from 200 samples.

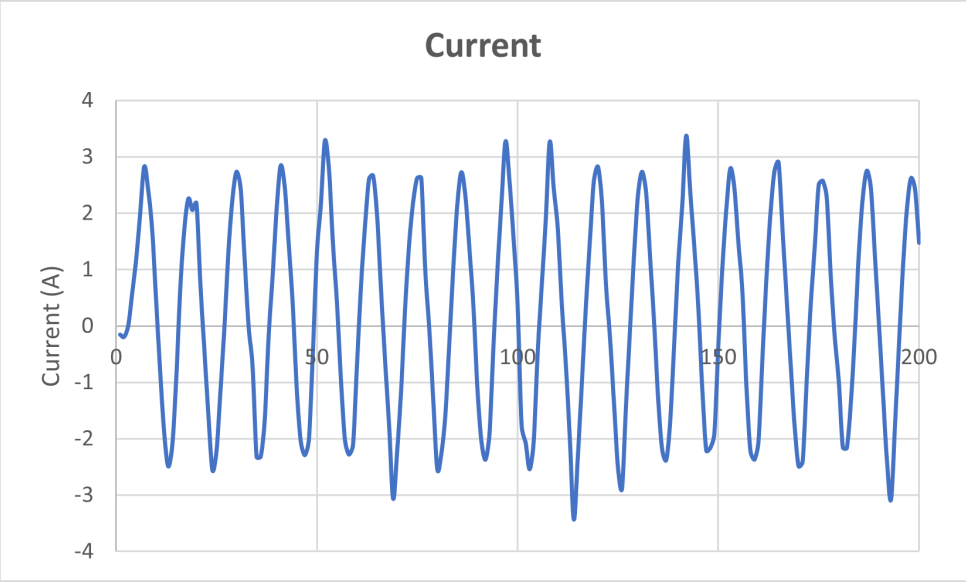


Figure 4.9: Graphical representation of the electric current inserted into the grid in a photovoltaic inverter.

Considering Figures 4.8 and 4.9, it can be seen that the voltage values reach peak values in the positive semicycle of 350V and in the negative semicycle of -300V. The peak current values are close to 3A and -3A in the positive and negative semicycles, respectively. Finally, the active power values are over 1000 W peak positive.

As mentioned previously, the power readings are processed by Node-RED, as per Figure 3.7 and stored in InfluxDB. Figure 4.10 presents power data obtained from the system connected to the Photovoltaic (PV) inverter.

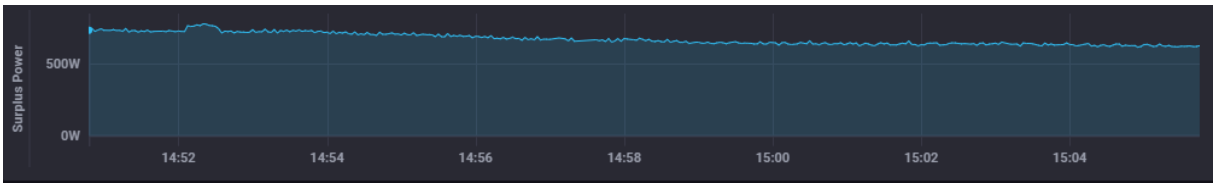


Figure 4.10: Power data read from PV inverter and stored in InfluxDB.

The same readings stored in the database can also be viewed in Grafana, as presented by Figure 4.11.

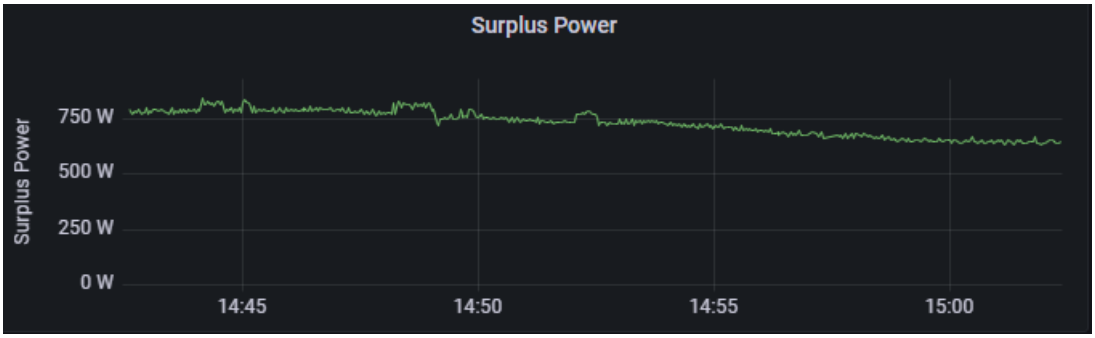


Figure 4.11: Power data read from PV inverter and viewed at Grafana.

Finally, the Home Assistant interface can also be used for data monitoring, as shown by Figure 4.12.

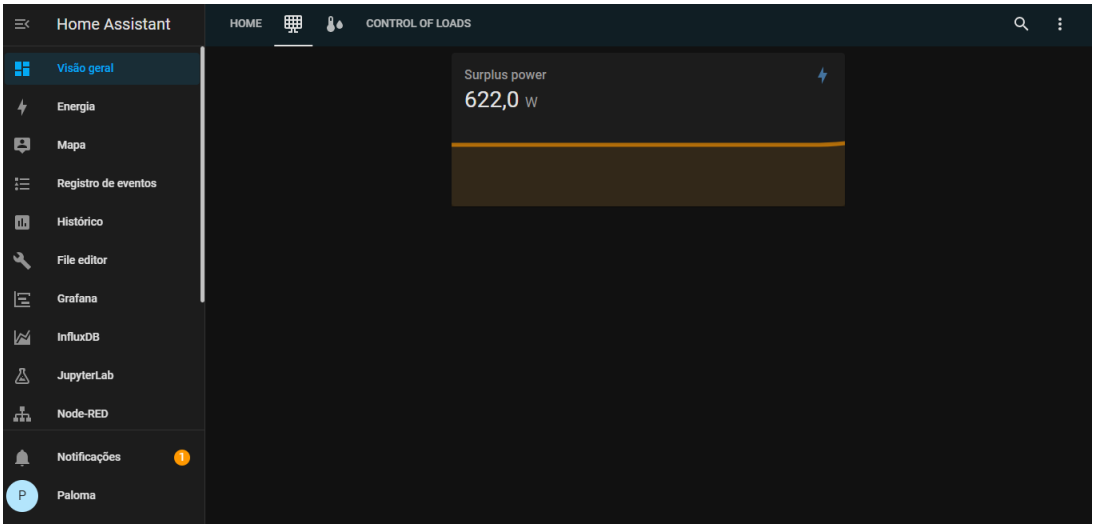


Figure 4.12: Power data read from PV inverter and viewed at Home Assistant interface.

The results obtained with the measurements satisfy the theoretical concepts and effectively demonstrate the functionality and evolution possibilities of the prototype.

## 4.2 Analysis of Machine Learning

In this topic, the results obtained with experiments performed for surplus energy prediction by means of linear regression and the corresponding analyses will be presented.

### 4.2.1 Forecast with hours

For the forecast of surplus energy using the intervals of 1h, 6h, and 12h in the creation of the model for the 1st, 7th, 14th, and 21st days of the first month of the year, the insertion of 1 to 6 features as predictor values of X for the construction of the model was considered. The predictions made with data from the 1st day will be graphically presented to improve the understanding of the application of this analysis. As this study was extended to the other days, the errors between actual values and predicted values in all these days will be discussed next.

Considering the first hour of the 1st day, it is to be expected that the model will not be successful, as there are 4 readings at night. Thus, Figure 4.13 evidently demonstrates the failure of a model created considering this data.

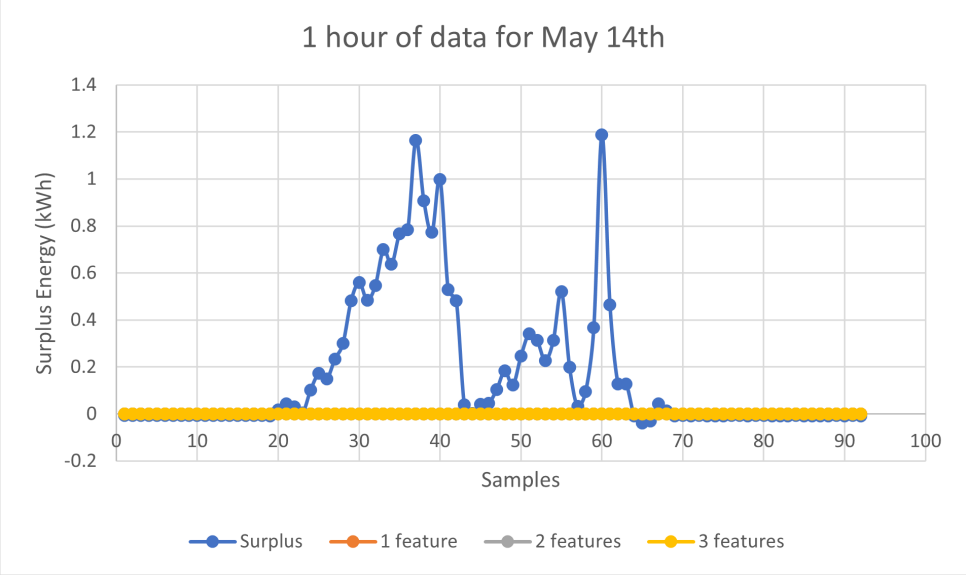


Figure 4.13: Graphical representation of forecasts using 1 hour of day 14 to forecast the rest of the day. Due to the proximity of the values obtained with the models that use 1 to 3 features as X predictor values, the graphical visualization of all the results was impaired.

It is observed that there are only 3 options of features in X. The reason for this is that 1 hour of data was used to create the models. This period has only 4 data, which allows the use of up to 3 features for X. And in all cases, with 1, 2, or 3 features, the predicted

values are null.

In addition, considering the first 6 hours of May 14th to create a regression model that can have 1 to 6 features for $X$, the forecast for the remainder of May 14th was developed and the results are presented in the graph of Figure E.1, in the appendix E. It is possible to observe that the forecast models developed with 1 to 6 features present very similar results.

With regard to the use of these same models with data from the first 6 hours of May 14 to forecast the entire day of May 15, the forecast results are shown in Figure E.2, in the Appendix E. The prediction results using 1 to 6 features are again similar. Although the difference is minimal compared to the other numbers of features, it is possible to note that using only 1 feature for $X$, i.e., only $X1$, the results are closer to the real values.

Regarding the predictive analysis using the first 12 hours of May 14 to predict the rest of that same day, considering 1 to 6 features, Figure E.3 presents the results obtained, in the Appendix E. It can be seen that predictions made with only 1 feature have a more relevant performance than the others.

Considering the forecast for the following day, using the same models based on the first 12 hours of day 14, the results shown in Figure E.4 are obtained, in the Appendix E. Again, the performance shown in the prediction that uses only 1 feature of $X$ for the linear regression model is better evaluated.

Also, considering that regression models were created with 1h, 6h and 12h data of May 21st, May 28th and June 4th, that represent the 7th , 14th and 21st days of the first month of the year under analysis, and using tests from 1 to 6 features as X predictor values, in what follows the graphic comparisons of the mean absolute error values in each case will be presented. As mentioned, the ineffectiveness of using 1h of the day to predict the surplus energy of the same day was considered and therefore only the errors related to 6h and 12h models will be compared. Figure 4.14 presents the mean absolute errors in regression models created with 6-hour data from May 14th both for predicting the rest of May 14th and, in another case, for predicting the next day, May 15th. The study is based on tests using 1 to 6 features for the creation of the models and presents the performance

of each from the mean error values.

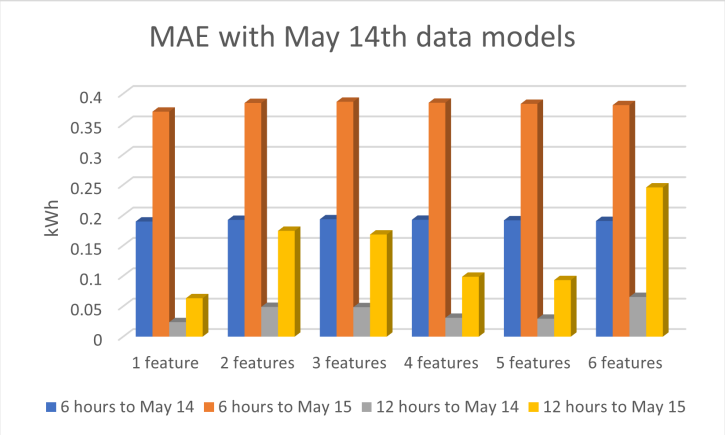


Figure 4.14: Mean absolute error in forecasts with models created from hours data on May 14th.

It can be seen that the lowest error values are set by 12-hour models of May 14th to make forecasts for the same day. Especially the model that uses 1 feature as $X$ predictor value, its error is significantly lower than the others. Moreover, the 12-hour models for forecasting data for the day after the 14th have relevant performance, even more so when compared to the 6-hour models. This same analysis of the mean absolute error for the 7th day of the month under consideration, i.e., for May 21st, was applied and the results are presented in Figure 4.15.

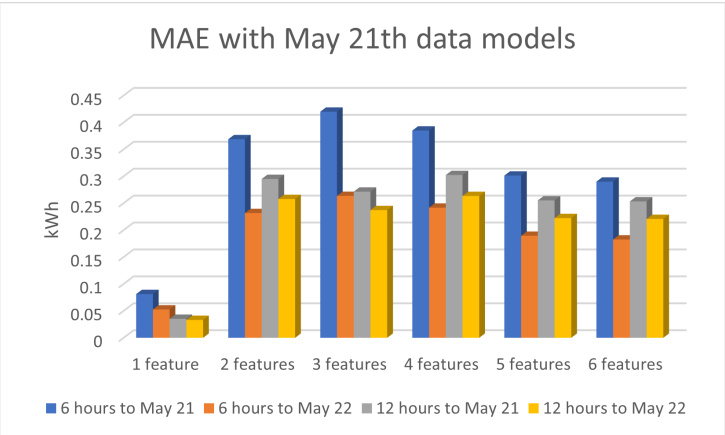


Figure 4.15: Mean absolute error in forecasts with models created from hours data on May 21st.

It can be concluded that the 6-hour models for forecasting the day after May 21th highlighted in relation to the value of the mean absolute error, especially with 1 feature. Then, the 12-hour models also developed for the May 22nd forecast have relevance in the mean error values, with higher emphasis on the model that uses 1 feature for $X$. Considering the 14th day for model creation, i.e., May 28th, Figure 4.16 presents the results obtained.

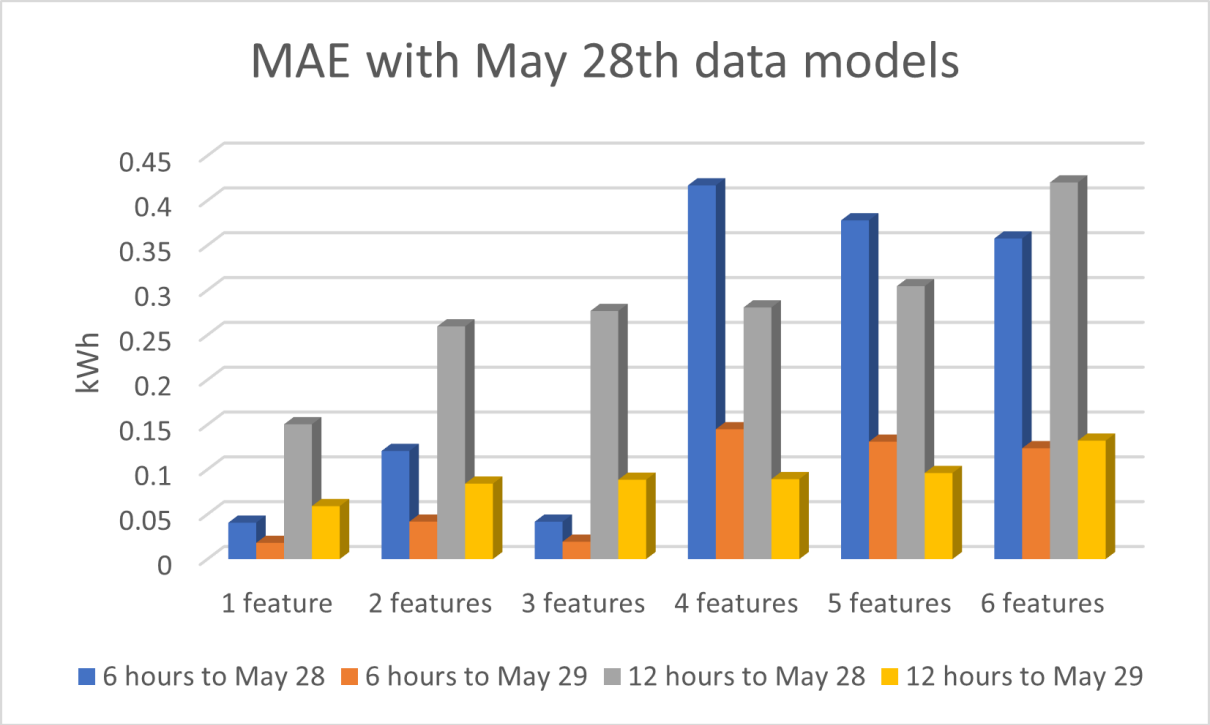


Figure 4.16: Mean absolute error in forecasts with models created from hours data on May 28th.

Again, the models that used 6 hours and 12 hours on May 28th for the next day's forecast had lower mean absolute errors, especially the 1 to 3 feature models for $X$. Finally, this study was done for the 21st day of the month in consideration, which corresponds to June 4th. The results are presented in the graph in Figure 4.17.

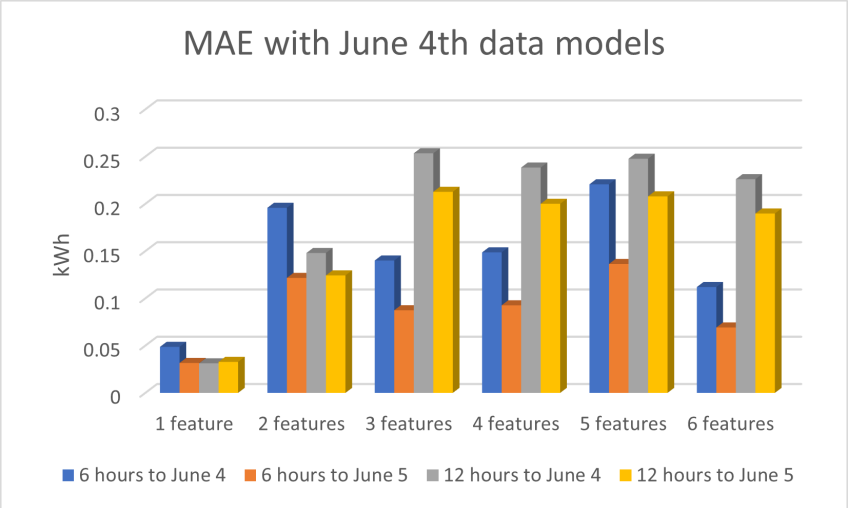


Figure 4.17: Mean absolute error in forecasts with models created from hours data on June 4th.

In this case, the focus will be given to the performance of the June 4 6-hour models predicting June 5th data. The mean absolute error values were the lowest compared to the other analyses, with the lowest error being obtained by the model using 1 feature as X predictor value.

After these analyses, the model of the first 12 hours of May 14th was adopted to predict the next 31 days, i.e., between May 15th and June 14th, 2016. The accuracy of these models for different features as $X$ predictor values are shown in Figure 4.18.

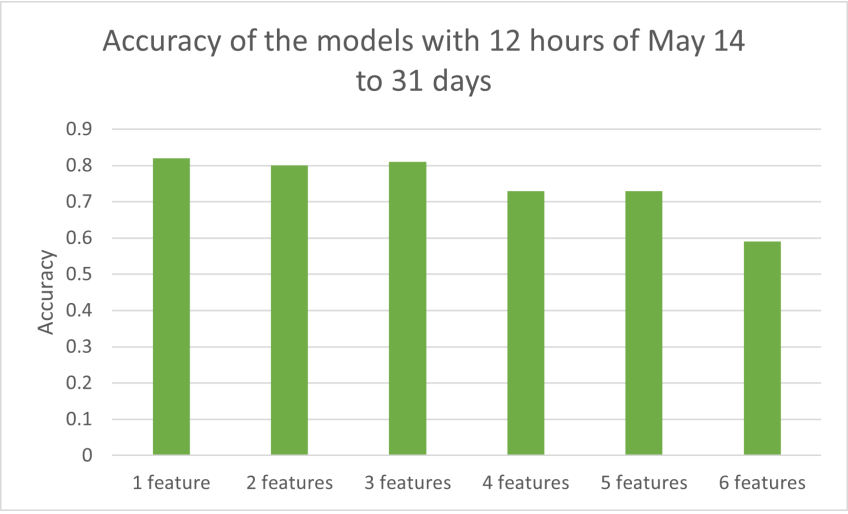


Figure 4.18: Accuracy of the regression models created from the first 12 hours of May 14.

As noted in previous predictive analyses and in agreement with the results obtained in the Figure 4.18 graph, the May 14th 12-hour model has a more relevant performance when 1 feature is defined as the predictor value X. It is possible to observe that the accuracy, in this case, is around 0.83. After evaluating the accuracy of the models and applying the prediction tests for the 31 days following May 14th, it is possible to verify the error values, which correspond to the difference of the actual value and the predicted value. With the error values, the results of mean absolute error and median absolute error are obtained, as shown in Figure 4.19.

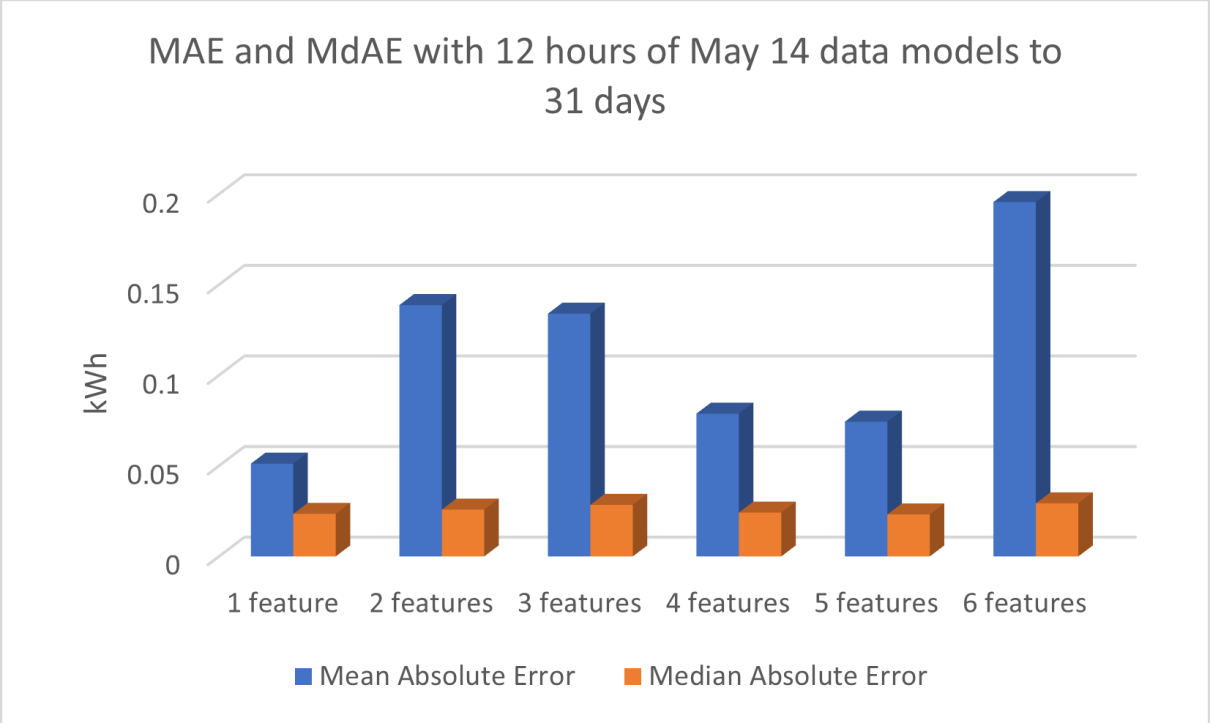


Figure 4.19: Mean absolute error and median absolute error values considering forecasts for the 31 days following May 14th.

With a special focus on the mean absolute error, the model based on the first 12 hours of May 14th and using 1 feature presents the lowest MAE value for the following 31 days, being approximately 0.05 kWh. In the same sense, the forecast error variation, defined by the measurement error variance, over this period is shown in Figure 4.20.

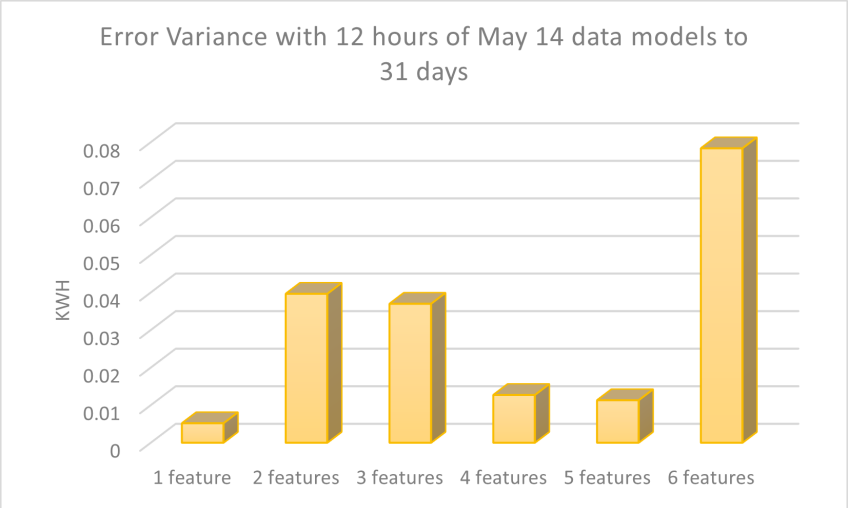


Figure 4.20: Error variance values considering forecasts for the 31 days following May 14.

The lowest error variation also occurs for the model based on 1 feature for $X$, being less than 0.01 kWh. Finally, the measurements obtained for the explained variance score, demonstrating the ability of the model to explain error variations, are presented in Figure 4.21.

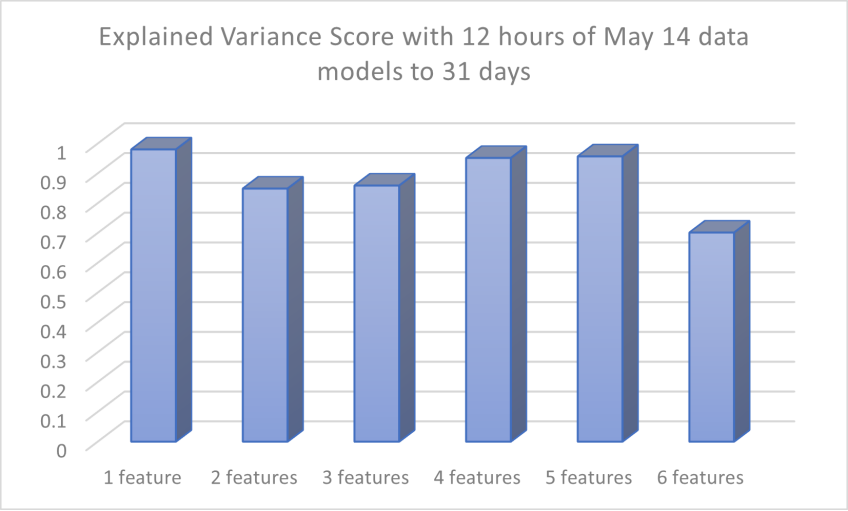


Figure 4.21: Explained Variance Score values considering forecasts for the 31 days following May 14.

Understanding that the closer to 1, the better the explained variance score is, for the 12-hour model with 1 feature this parameter has better performance, with a value above

0.9, which means the high capacity of the model that uses 1 feature for X to explain the variation in the data. Concluding the investigation applied to the use of hours of energy generation data in kWh, considering samples of 15 minutes, for the creation of forecast models through linear regression, it was possible to observe a relevant performance in models that used 6 hours and 12 initial hours of the data day and 1 feature for $X$. In the context of using the 12-hour model from May 14 for the following 31 days, also with 1 feature for $X$, the forecast efficiency obtained, with error, it is possible to highlight the MAE of approximately 0.05 kWh.

### 4.2.2 Forecasts from one day to the next

As mentioned in Chapter 3, the first analysis using data from one day to predict the next was with May 14 to predict May 15. Figure 4.22 presents the predicted data of the 15th compared to the actual data, called *Surplus*.

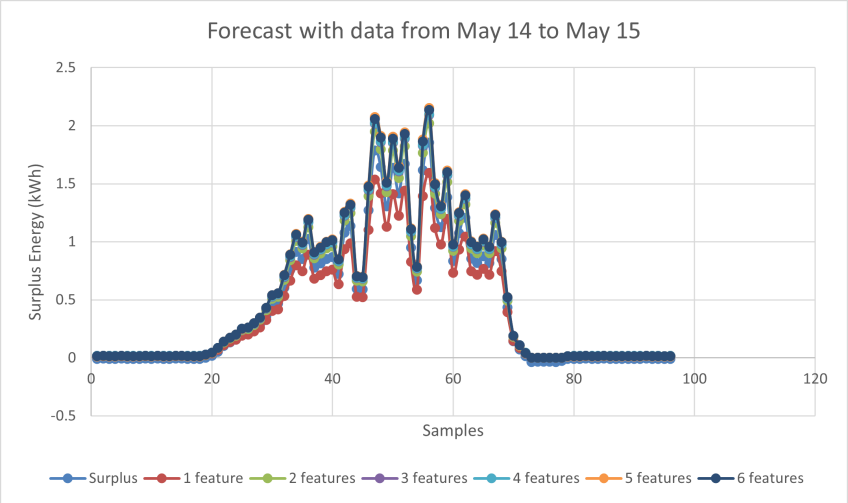


Figure 4.22: Forecast for May 15 based on May 14 data.

It can be seen that the built models with the surplus energy data from May 14 and using 2 features as $X$ predictor values perform a prediction closer to the actual values. To facilitate this visualization, Table 4.1, presents the values of accuracy, mean absolute error, median absolute error, explained variance score and error variance.

Table 4.1: Model created on May 14 / Forecast for May 15

| **Features** | Accuracy | MAE | MdAE | Explained Variance Score | Error Variance |
|---|---|---|---|---|---|
| **1 feature** | 0.69 | 0.0686 | 0.0247 | 0.976 | 0.00714 |
| **2 features** | 0.59 | 0.0587 | 0.0351 | 0.994 | 0.00173 |
| **3 features** | 0.54 | 0.0765 | 0.0386 | 0.987 | 0.00392 |
| **4 features** | 0.54 | 0.0794 | 0.0403 | 0.986 | 0.00419 |
| **5 features** | 0.48 | 0.0885 | 0.0387 | 0.978 | 0.00670 |
| **6 features** | 0.53 | 0.0870 | 0.0413 | 0.981 | 0.00571 |

Although the accuracy of the model with 2 features for $X$ is lower than with 1 feature, the mean absolute error value is 0.0587 kWh, the lowest value among the other models. The error variance, in this case, is also lower than in the other models. However, it is still important to note that the accuracy of all models does not correspond to significantly high values, which means that these models may not apply for all new data. Regarding the models developed with information from May 29 to predict May 30, Figure 4.23 graphically shows the predicted values as well as the actual values.

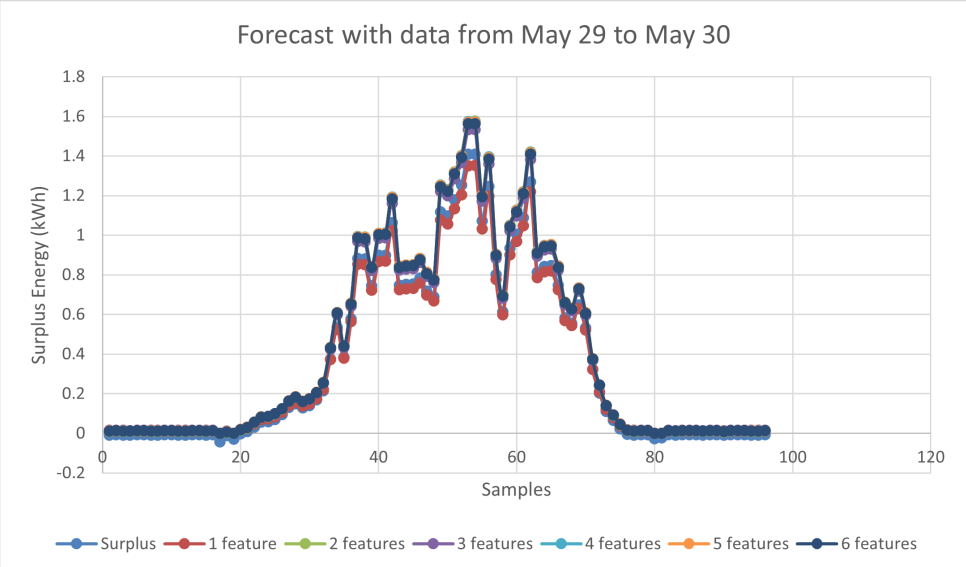


Figure 4.23: Forecast for May 30 based on May 29 data.

As seen in the graph in Figure 4.23, the predicted values for May 30 with the model

based on the May 29 data using 1 feature for $X$ have a closer approximation to the actual values. Again, to aid visualization, Table 4.2 presents parameters calculated in this study.

Table 4.2: Model created on May 29 / Forecast for May 30

| **Features** | Accuracy | MAE | MdAE | Explained Variance Score | Error Variance |
|---|---|---|---|---|---|
| **1 feature** | 0.88 | 0.0211 | 0.0171 | 0.997 | 0.000576 |
| **2 features** | 0.91 | 0.0564 | 0.0299 | 0.989 | 0.00211 |
| **3 features** | 0.93 | 0.0490 | 0.0305 | 0.995 | 0.000982 |
| **4 features** | 0.89 | 0.0569 | 0.0304 | 0.990 | 0.00205 |
| **5 features** | 0.88 | 0.0577 | 0.0319 | 0.990 | 0.00189 |
| **6 features** | 0.86 | 0.0548 | 0.0305 | 0.991 | 0.00171 |

Analyzing Table 4.2, the model with 1 feature has a mean absolute error of 0.0211 kWh, the lowest among the other error values resulting from models with more features for $X$. In addition, the error variance for this case is also lower. The accuracy of this model is significant, although not the highest among the other models, at around 0.88.

For the June 4 surplus generation forecast from June 3 data, Figure 4.24 below graphically presents the results obtained.

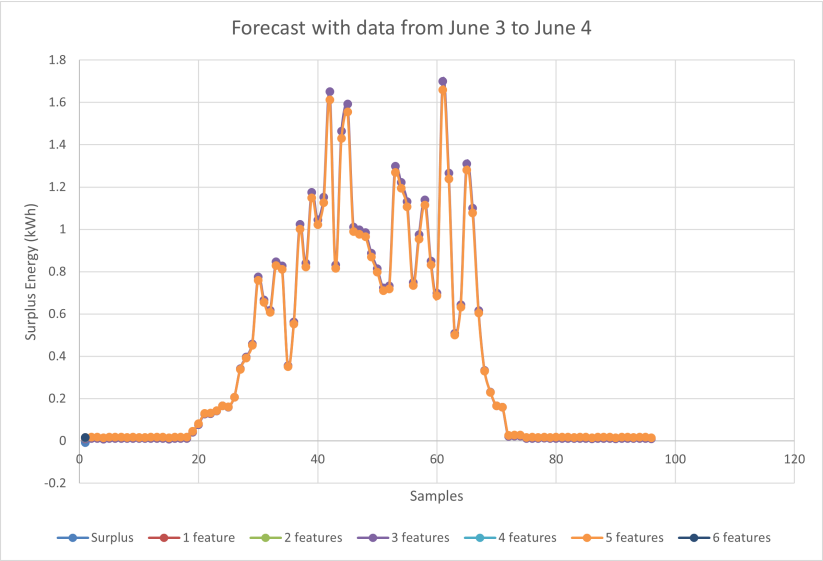


Figure 4.24: Forecast for June 4th based on June 3 data. Due to the proximity of the values obtained with the models that use 1 to 6 features as X predictor values, the graphical visualization of all the results was impaired.

The predicted data in all models is close and makes it difficult to see graphically which model is best for this case. Thus, to conclude this analysis, Table 4.3 presents the parameters obtained in this context.

Table 4.3: Model created on June 3 / Forecast for June 4

| **Features** | Accuracy | MAE | MdAE | Explained Variance Score | Error Variance |
|---|---|---|---|---|---|
| **1 feature** | 0.91 | 0.0139 | 0.0178 | 0.999 | 0.000192 |
| **2 features** | 0.91 | 0.0395 | 0.0264 | 0.997 | 0.000560 |
| **3 features** | 0.91 | 0.0516 | 0.0299 | 0.993 | 0.00153 |
| **4 features** | 0.92 | 0.0694 | 0.0388 | 0.985 | 0.00308 |
| **5 features** | 0.92 | 0.0460 | 0.0317 | 0.997 | 0.000670 |
| **6 features** | 0.92 | 0.0458 | 0.0316 | 0.997 | 0.000658 |

According to Table 4.3, the lowest MAE value is for the model with 1 feature for $X$, with a value of 0.0139 kWh. Not only the MAE, but also the error variance is lower, and the accuracy is significantly high, reaching a value of 0.91.

Finally, to complete the investigation of the use of data from one day to forecast surplus generation for the next, there is an analysis performed with information from June 13 to forecast the 14. Figure 4.25 shows the obtained results.

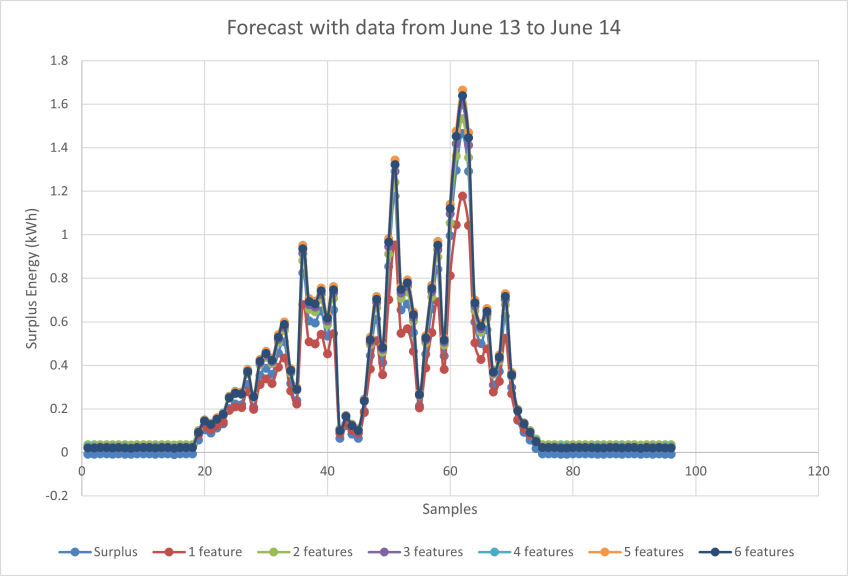


Figure 4.25: Forecast for June 14 based on June 13 data.

From the graph, it is possible to note that the closest approximation of the forecast to the actual readings is with the data obtained from the model that uses 2 features for $X$. Table 4.4 shows in more detail the criteria measured for this case.

Table 4.4: Model created on June 13 / Forecast for June 14

| **Features** | Accuracy | MAE | MdAE | Explained Variance Score | Error Variance |
|---|---|---|---|---|---|
| **1 feature** | 0.74 | 0.0574 | 0.0380 | 0.951 | 0.00573 |
| **2 features** | 0.73 | 0.0476 | 0.0450 | 0.9997 | 3.013E-05 |
| **3 features** | 0.80 | 0.0513 | 0.0404 | 0.995 | 0.000534 |
| **4 features** | 0.78 | 0.0633 | 0.0452 | 0.988 | 0.00145 |
| **5 features** | 0.8 | 0.0618 | 0.0433 | 0.987 | 0.00152 |
| **6 features** | 0.81 | 0.0552 | 0.0394 | 0.991 | 0.00111 |

As can be seen, the lowest mean absolute error is around 0.0476 kWh, and is the result of applying the model with 2 features to X, which also generates the lowest error variance. The accuracy of this model, on the other hand, although significant, around 0.73, is the lowest among the other models presented in the table.

In general, it is worth mentioning that the models created from May 14th to May 15th have lower predictive efficiency when compared to the other analyses. However, it is possible to note forecast results very close to the real ones, which were highlighted throughout this topic. The following topics will introduce models that use multi-day data to perform forecasting.

### 4.2.3 Using 5 days for forecast

The surplus generation measures from May 14th to May 18th, 2016, were used to initially forecast the data from May 19th, using analyzes of 1 to 6 features for $X$. Figure 4.26 presents the forecast results compared to the real values.

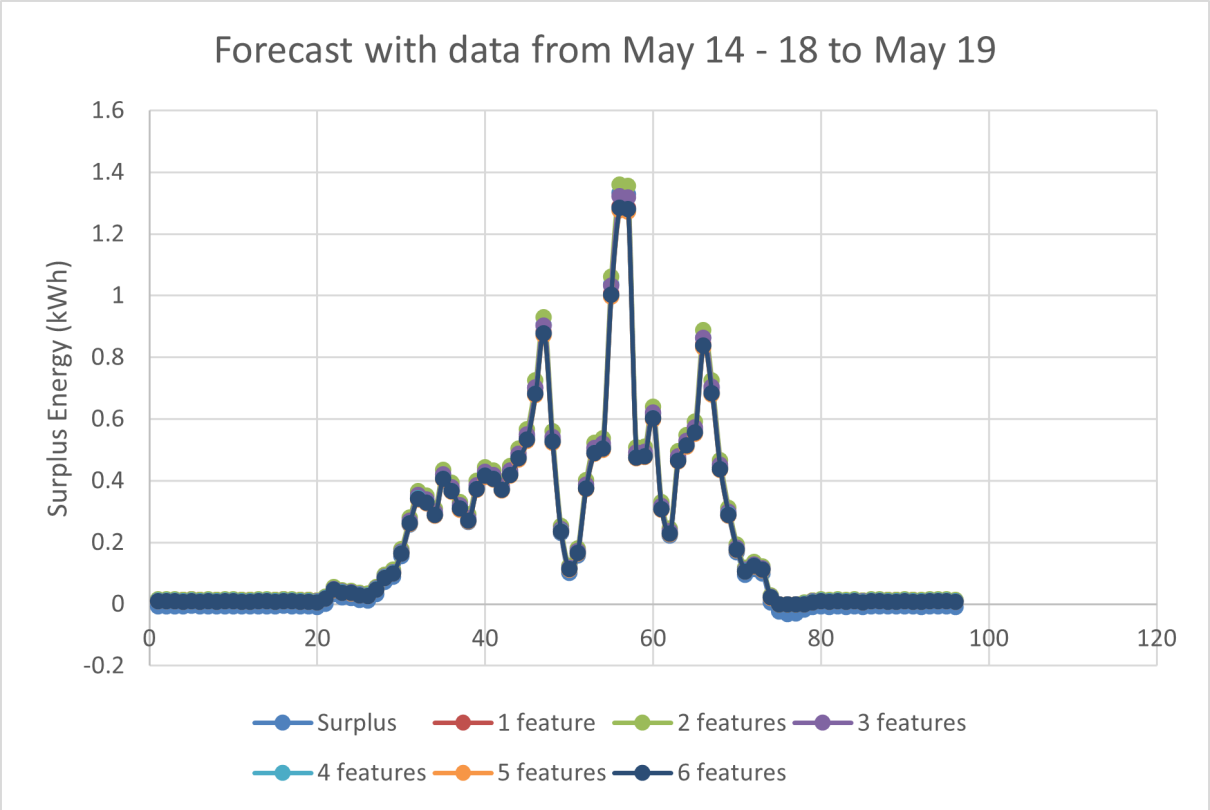


Figure 4.26: Forecast for May 19 based on May 14 - 18 data.

From the graph in Figure 4.26, it is possible to observe that the behavior of the data forecast for May 19th is similar and follows the pattern of the actual data. However, to facilitate the analysis of the results, Table 4.5 presents parameters obtained from the developed models and predicted data.

Table 4.5: Model created on May 14 - 18 / Forecast for May 19

| **Features** | Accuracy | MAE | MdAE | Explained Variance Score | Error Variance |
|---|---|---|---|---|---|
| 1 feature | 0.78 | 0.0161 | 0.0207 | 0.998 | 0.000211 |
| 2 features | 0.77 | 0.0236 | 0.0229 | 0.99997 | 2.60E-06 |
| 3 features | 0.83 | 0.0147 | 0.0177 | 0.999 | 4.82E-05 |
| 4 features | 0.83 | 0.0146 | 0.0176 | 0.997 | 0.000247 |
| 5 features | 0.82 | 0.0142 | 0.0157 | 0.997 | 0.000267 |
| 6 features | 0.83 | 0.0136 | 0.0161 | 0.998 | 0.000216 |

From the table, it is possible to note that the lowest MAE is around 0.0136 kWh, referring to the model that uses 6 features for $X$. This same model is among those with the highest accuracy, around 0.83. However, the lower error variance is related to the model with 2 features, the same one that has the highest value of explained variance score.

Adopting an analysis with 3 features, which also showed a high performance in the previous forecast, and applying the model developed with data from these 5 days for the rest of the year, predicting the same day 19 in 11 months, the mean absolute error and median absolute error values are obtained and presented in Figure 4.27.

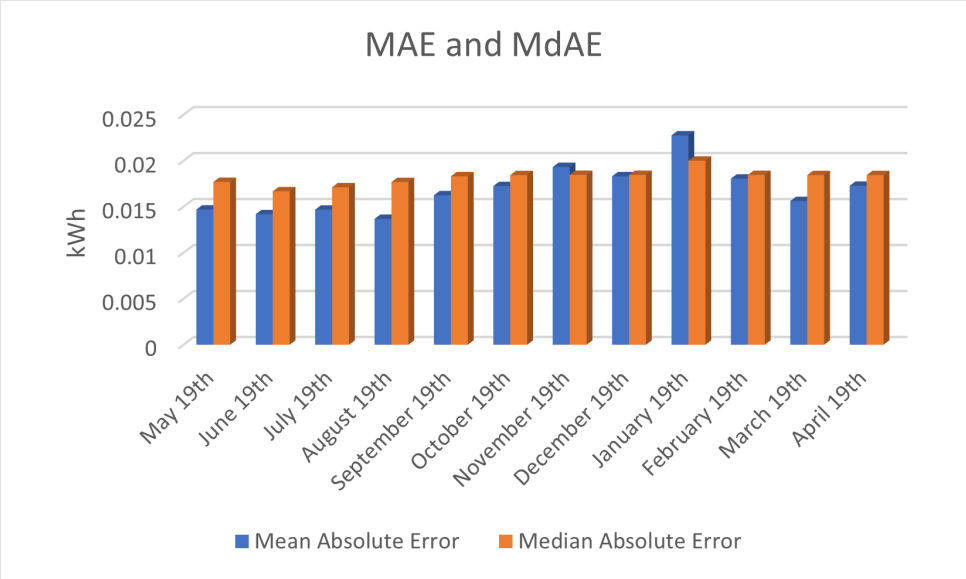


Figure 4.27: Mean absolute error and median absolute error values considering forecasts for the same day 19th in 11 months based on May 14th-18th data.

It is observed that the MAE and MdAE parameters are not so discrepant throughout the year, being below 0.02 kWh in most months. It is noted that a significant increase in these values occurred on January 19, but without a relevant difference.

With regard to the error variance in the forecasts made, Figure 4.28 shows the error variance of the forecast data for the 19th of each month for 11 months.

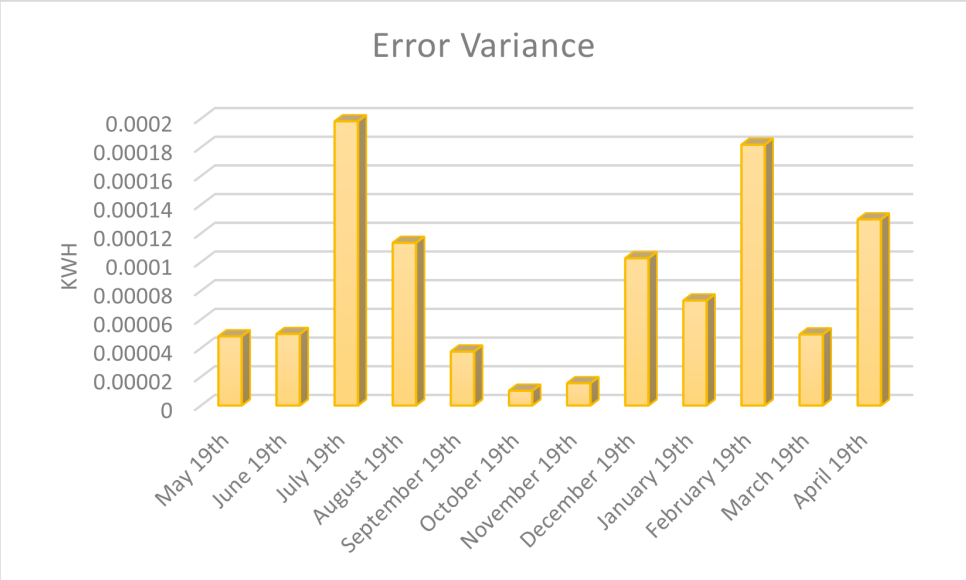


Figure 4.28: Error variance values considering forecasts for the same day 19th in 11 months based on May 14th-18th data.

It is possible to observe that the error variance is higher on July 19th, being approximately 0.0002 kWh and has its lowest value on October 19th, with a value close to 0.00001 kWh. Regarding the explained variance score, Figure 4.29 presents the results obtained for this study.

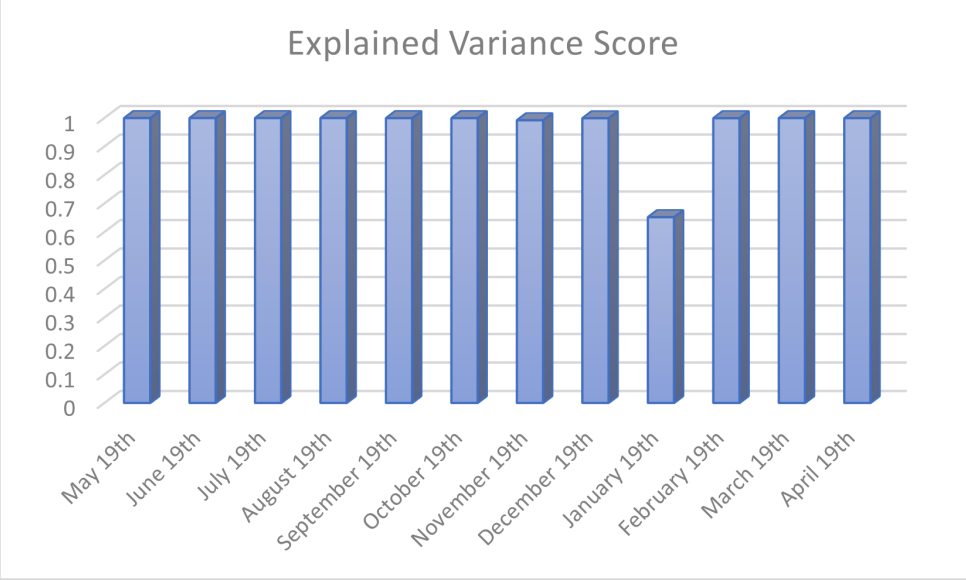


Figure 4.29: Explained Variance Score values considering forecasts for the same day 19th in 11 months based on May 14th-18th data.

With the exception of the 19th of January, on all the other 19th of the other months, the explained variance score has a significantly high value, which implies the effectiveness of the model to explain the variation in the data.

A significant performance was noted in the models based on May 14-18 for the forecast of certain days throughout the year. The next topic will use a wider range of data to analyze forecasting efficiency.

### 4.2.4 Using 10 days for forecast

Considering the 10-day based forecast study, we used May 14-23, 2016 to forecast May 24, using 1 to 6 features for X. In this regard, Figure 4.30 presents the results obtained.

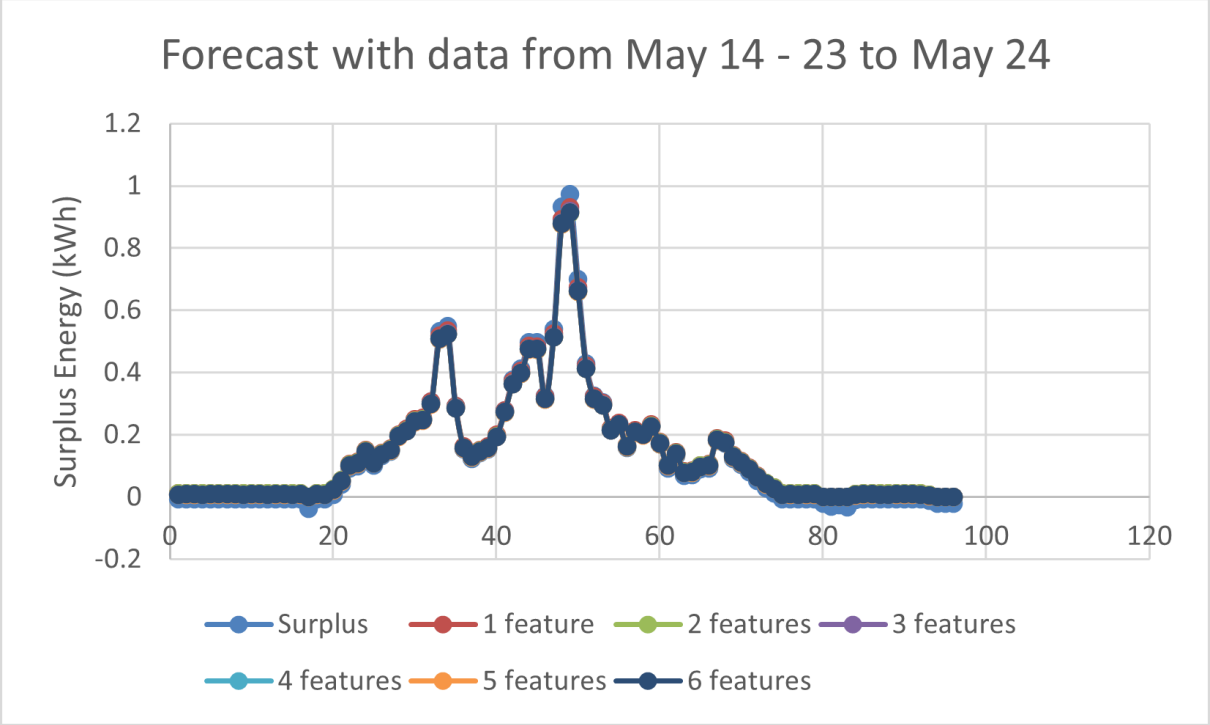


Figure 4.30: Forecast for May 24 based on May 14 - 23 data.

It can be seen that the forecasts do not differ significantly from the real values. To extend the analysis of the forecast efficiency, Table 4.6 shows parameters calculated for this case.

Table 4.6: Model created on May 14 - 23 / Forecast for May 24

| Features | Accuracy | MAE | MdAE | Explained Variance Score | Error Variance |
|---|---|---|---|---|---|
| 1 feature | 0.90 | 0.0151 | 0.0163 | 0.996 | 0.000180 |
| 2 features | 0.90 | 0.0156 | 0.0191 | 0.993 | 0.000289 |
| 3 features | 0.91 | 0.0131 | 0.0149 | 0.994 | 0.000235 |
| 4 features | 0.91 | 0.0132 | 0.0147 | 0.994 | 0.000255 |
| 5 features | 0.91 | 0.0132 | 0.0145 | 0.994 | 0.000260 |
| 6 features | 0.91 | 0.0138 | 0.0160 | 0.994 | 0.000258 |

As shown in Table 4.6, it can be seen that in all the models created, the accuracy presents high values, above 0.90. Regarding the MAE, the values are close, being lower with models that include 3 to 6 features. The explained variance score is also high for both models, around 0.99. The variance error is lower for the model with 1 feature but does not differ relevantly from the others.

Considering the significant results obtained, the model with 3 features was again adopted to predict all days 24 over 11 months. Figure 4.31 presents the MAE and median absolute error values obtained.

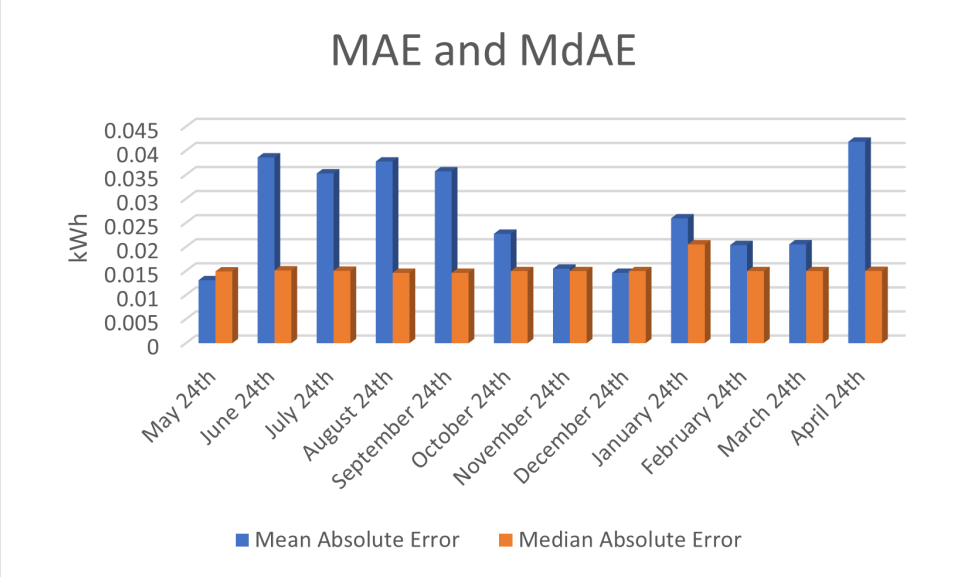


Figure 4.31: Mean absolute error and median absolute error values considering forecasts for the same day 24th in 11 months based on May 14th-23rd data.

It can be seen that the MAE was highest on the 24th day of the months between the months of June and September, and also in April 24. The MAE, in most months of the year was below 0.04 kWh, being half this value or less on the 24th day of the months of May, November, December, February and March. Regarding error variance, Figure 4.32 shows the values obtained for all days 24 over 11 months.

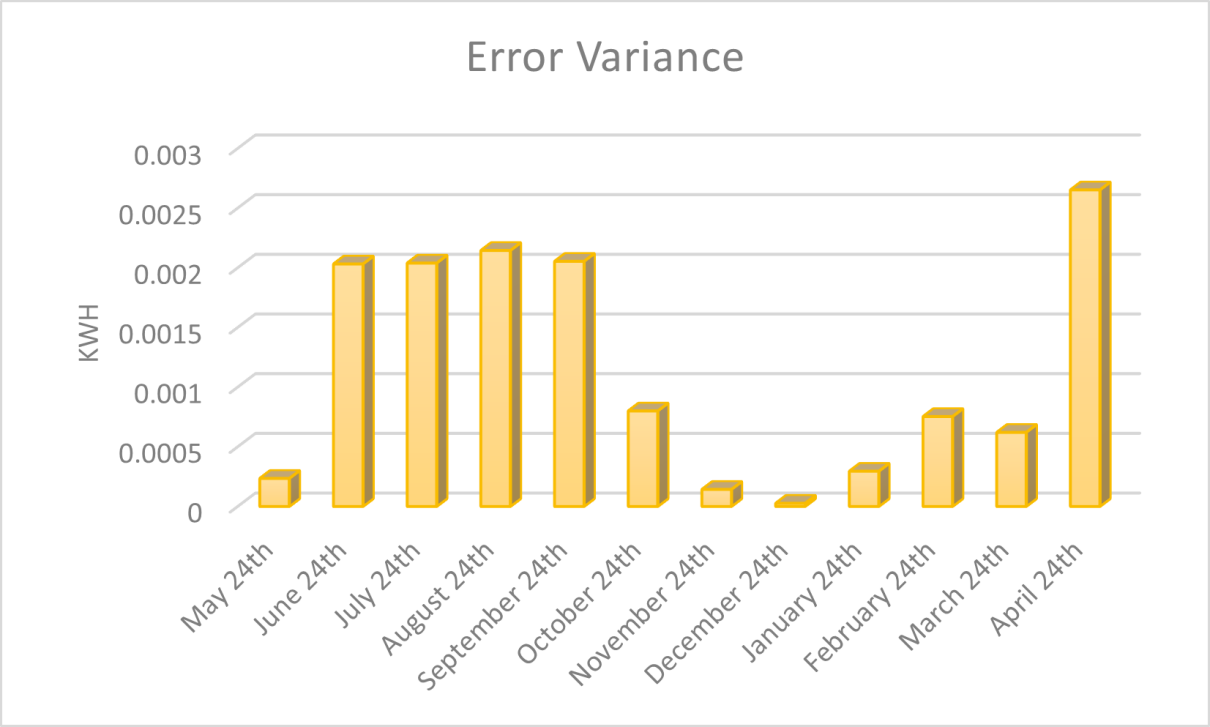


Figure 4.32: Error variance values considering forecasts for the same day 24th in 11 months based on May 14th-23rd data.

The error variance is notably higher on the 24th day of the months between June and September, as well as on the 24th day of April, the latter reaching a value above 0.0025 kWh. It has its lowest value on December 24th. Finally, analyzing the explained variance score, Figure 4.33 shows the values obtained.

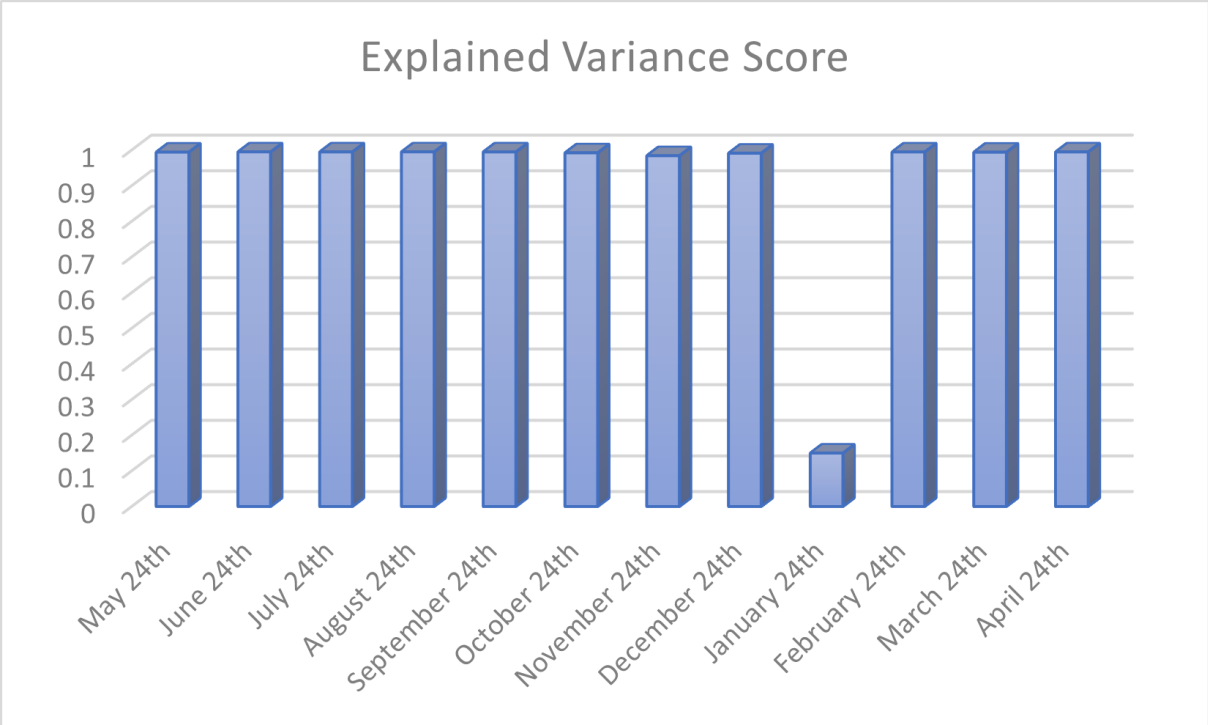


Figure 4.33: Explained Variance Score values considering forecasts for the same day 24th in 11 months based on May 14th-23rd data.

With the exception of January 24, which presented a low explained variance score, all other days 24 of the other months obtained values above 0.90 for this parameter, which means that the developed models can efficiently explain the data variation.

### 4.2.5 Using 30 days for forecast

As a last analysis, 30-day surplus generation readings from May 14 to June 13, 2016 were used to perform the surplus generation forecast for the 31st day, i.e., June 14. Figure 4.34 presents the forecast results obtained with models using 1 to 6 features for $X$, compared to the actual values, called *Surplus*.

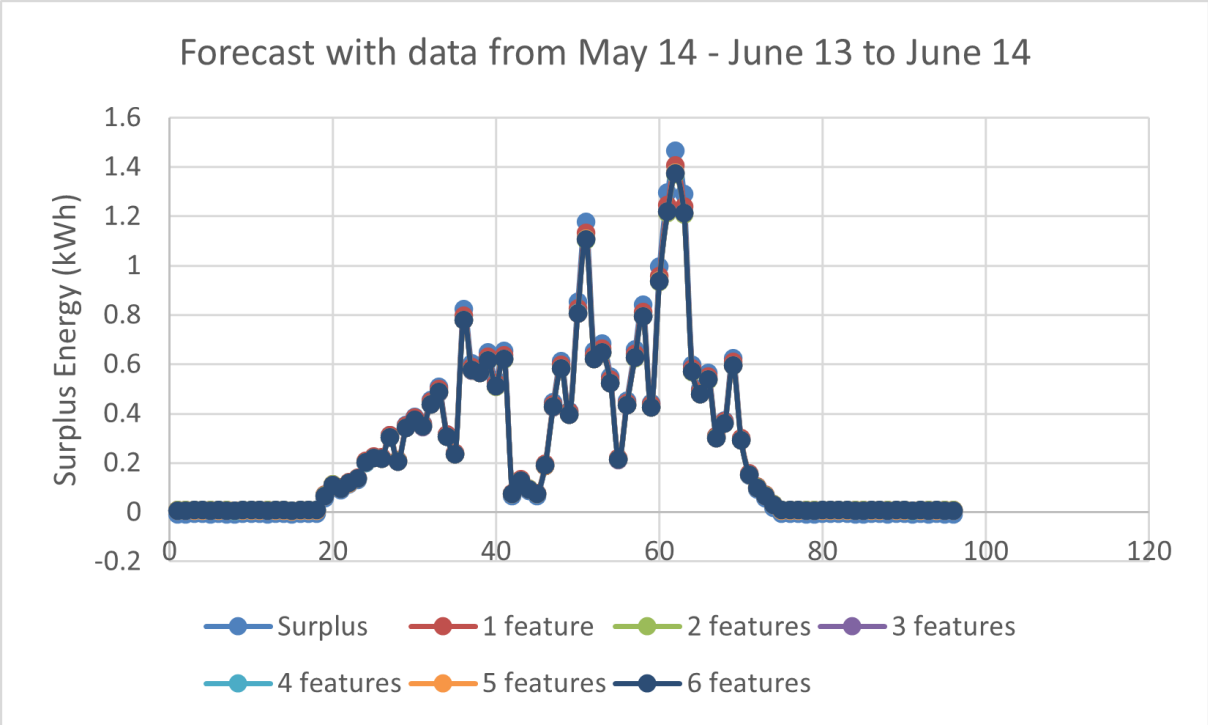


Figure 4.34: Forecast for June 14 based on May 14 - June 13 data.

From the graph, it can be seen that the model has results closer to the real ones when using 1 feature for $X$. However, to extend the analysis of the results, Table 4.7 presents the parameters calculated for this study.

Table 4.7: Model created on May 14 - June 13 / Forecast for June 14

| Features | Accuracy | MAE | MdAE | Explained Variance Score | Error Variance |
|---|---|---|---|---|---|
| 1 feature | 0.91 | 0.0152 | 0.0165 | 0.997 | 0.000324 |
| 2 features | 0.91 | 0.0202 | 0.0163 | 0.994 | 0.000680 |
| 3 features | 0.91 | 0.0185 | 0.0134 | 0.995 | 0.000584 |
| 4 features | 0.91 | 0.0183 | 0.0134 | 0.995 | 0.000569 |
| 5 features | 0.91 | 0.0184 | 0.0137 | 0.995 | 0.000577 |
| 6 features | 0.91 | 0.0189 | 0.0147 | 0.995 | 0.000604 |

Based on the parameter values presented, the accuracy values are significantly high for models using 1 to 6 features, equivalent to 0.91. The model that presents the lowest

mean absolute error is the one that uses 1 feature, being approximately 0.015 kWh, which explains the closer approximation with the real values seen in the graph of Figure 4.34. Moreover, the lowest error variance value also occurs with the model that uses 1 feature for $X$, being approximately 0.0003 kWh. However, the other values for the latter show no significant difference.

Considering the model with 1 feature for forecasting data on June 14, 2016, Figure 4.35 shows the linear fit obtained from the straight line expression that uses the values of $X1$ and the angular and linear coefficients of the model.

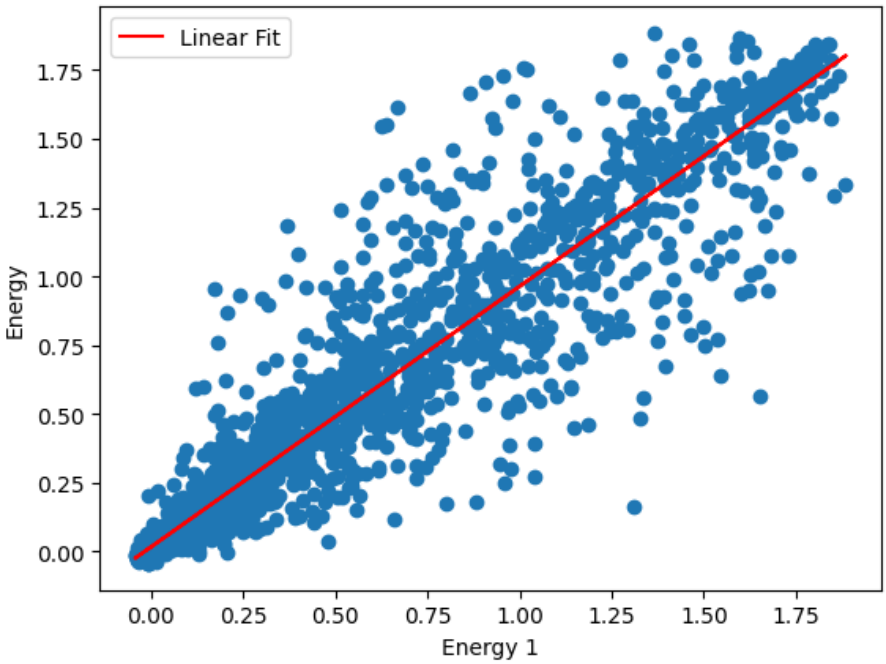


Figure 4.35: Linear Fit of the linear regression model using 1 feature.

For the extension of the analyses, one could study the use of other months of the year under consideration in the dataset for forecasting experiments. However, the aim of this study was to understand how much data was needed to have a robust forecast model and when this model should be updated. It is noted that it is possible to build an efficient forecast model with hours or with days. However, the question of updating the model still needs further study.

# Chapter 5

# Conclusion and Future Work

Based on all the work presented, the use of surplus energy generated for residential applications is a solution that traces a path different from conventional ones.Together with this proposal, the objective was to develop a system for monitoring and controlling surplus energy data and to participate in the system by making decisions when directing energy to a specific load. However, the proposal goes beyond viewing data and controlling energy consumption by loads remotely. The work seeks to create surplus prediction mechanisms to give the system the possibility of making decisions automatically, through Machine Learning.

In this sense, the first step of this project was to understand the operation of data processing, storage and monitoring platforms, which included Node-RED, InfluxDB, Grafana and Home Assistant. These tools, as well as the communication between them, were used in experiments to measure and monitor excess energy. For this purpose, an experimental hardware prototype was built, using the ESP32 as a microcontroller, to measure voltage and current in 3 situations: simulation of excess generation using a variable resistance; grid voltage and current reading to understand negative power behavior; reading voltage and current in a photovoltaic inverter connected to the grid to understand the behavior of positive or surplus power. The excess power values were monitored and stored in the database. It should be noted that control tests were carried out on a lamp and heater based on the ratio between excess power and power consumed by the load, the latter

informed by the user in the Home Assistant interface.

With regard to data analysis for the development of surplus power forecast, a dataset made available for use was used since the development time of the measurements would not allow the wide range of data that the dataset used had provided. Then, experiments were carried out to predict excess energy using the linear regression method. The challenges were aimed at understanding which is the best data interval that provides a robust forecasting model and what is the update time of this model. Especially in terms of the interval, several tests were performed, from hours to days, with 1 to 6 features for X, to understand the forecasting efficiencies. Models with 6 hours and 12 hours had significant forecast results for the next day, for example. 12-hour models also had considerable predictive efficiency for 31 days. On the other hand, the use of days to perform forecasting showed relevant results both for predicting data from later days and for predicting data from certain days in certain months. It is still possible to develop numerous analyzes to effectively achieve answers such as the forecast model update time.

Thus, to understand the context of future work, it is important to approach again the operation of the automatic decision-making system and the communication between its components, as well as what has already been applied and the future possibilities of evolution. As presented throughout the paper, the excess generation values are read by the built metering prototype and sent to the MQTT broker. The Node-RED processes the data and sends it to the database, both tools being implemented in the Home Assistant, to facilitate control and monitoring actions. Based on the surplus generation history stored in InfluxDB, the Python script accesses the history data and creates forecasting models to predict surplus energy. However, for the system to perform automatic decision-making, consumption data must be obtained, processed and stored in the database, as well as the surplus data. A Python script uses the historical energy usage to perform the consumption forecast. This step can be implemented as future work. In this way, based on the consumption and surplus generation forecast, the system will be able to make decisions about directing energy to the loads. In addition, further analysis on when to update the forecast models should be performed from experiments, in order to ensure

the efficient operation of the intelligent system. It is understood that intervals of hours for model generation or days can generate efficient forecast models depending on how long that model will be used for. This approach can also be better worked out in future developments.

Thus, this work is considered a starting point for further research and the development of new analyzes that will provide the proposed smart system with more robustness and efficiency in its decision-making.

# Bibliography


[1] R. O. Ibrahim, E. Tambo, D. Tsuanyo, and A. Nguedia-Nguedoung, "Modelling an artificial intelligence-based energy management for household in nigeria," *Engineering Letters*, vol. 30, pp. 1–12, 2022.

[2] M. H. Elkholy, T. Senjyu, M. E. Lotfy, A. Elgarhy, N. S. Ali, and T. S. Gaafar, "Design and implementation of a real-time smart home management system considering energy saving," *SUSTAINABILITY*, p. 14, 2022.

[3] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, "Security vulnerabilities of internet of things: A case study of the smart plug system," *IEEE INTERNET OF THINGS JOURNAL*, vol. 4, no. 6, 2017.

[4] M. Amer, A. Naaman, N. K. M'Sirdi, and A. M. El-Zonkoly, "Smart home energy management systems survey," in *International Conference on Renewable Energies for Developing Countries 2014*, 2014, pp. 167–173. DOI: 10.1109/REDEC.2014.7038551.

[5] J. Leitão, P. Gil, B. Ribeiro, and A. Cardoso, "A survey on home energy management," *IEEE Access*, vol. 8, pp. 5699–5722, 2020. DOI: 10.1109/ACCESS.2019.2963502.

[6] V. Murty and A. Kumar, "Multi-objective energy management in microgrids with hybrid energy sources and battery energy storage systems," *Protection and Control of Modern Power Systems*, vol. 5, no. 1, pp. 1–20, 2020.

[7] N. Mohan, *Electric power systems: a first course*. John Wiley and Sons, 2012, p. 132.

[8] P. S. Georgilakis and N. D. Hatziargyriou, "A review of power distribution planning in the modern power systems era: Models, methods and future research," *Electric Power Systems Research*, vol. 121, pp. 89–100, 2015, ISSN: 0378-7796. DOI: \url{https://doi.org/10.1016/j.epsr.2014.12.010}. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779614004490.

[9] ERSE, *Relatório da consulta pública da reformulação do rt do setor elétrico*, https://www.erse.pt/media/100l12wv/relatorio.pdf, Adapted from "Evolução das perdas nas redes distribuição de 1999 a 2019 e linha de tendência da evolução antes de 2011" by Paloma Greiciana de Souza Dias.

[10] E. R. dos Serviços Energéticos, *Diário da república, 2.a série - no 13*, https://files.dre.pt/2s/2019/01/013000000/0261902654.pdf, December, 2022.

[11] E. R. dos Serviços Energeticos, *Diário da república, 2.ª série - n° 163*, https://www.erse.pt/media/34vd1r4t/regulamento-785-2021_rt-se.pdf, December, 2022.

[12] ERSE, *Números e estatísticas/ eletricidade*, https://www.erse.pt/numeros-e-estatisticas/eletricidade/, December, 2022.

[13] PORDATA, *Consumidores de eletricidade: Total e por tipo de consumo*, https://www.pordata.pt/portugal/consumidores+de+eletricidade+total+e+por+tipo+de+consumo-1123, Adapted from "Consumidores de energia elétrica por tipo de consumo" by Paloma Greiciana de Souza Dias.

[14] PORDATA, *Consumo de energia elétrica: Total e por tipo de consumo*, https://www.pordata.pt/portugal/consumo+de+energia+eletrica+total+e+por+tipo+de+consumo-1124, Adapted from "Consumo de energia elétrica: total e por tipo de consumo" by Paloma Greiciana de Souza Dias.

[15] R. Passey, T. Spooner, I. MacGill, M. Watt, and K. Syngellakis, "The potential impacts of grid-connected distributed generation and how to address them: A review of technical and non-technical factors," *Energy Policy*, vol. 39, no. 10, pp. 6280–6290, 2011, Sustainability of biofuels, ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.

enpol.2011.07.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301421511005519.

[16] S. M. Ismael, S. H. Abdel Aleem, A. Y. Abdelaziz, and A. F. Zobaa, "State-of-the-art of hosting capacity in modern power systems with distributed generation," *Renewable Energy*, vol. 130, pp. 1002–1020, 2019, ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2018.07.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148118307936.

[17] C. E. d. S. Santiago, G. T. Campos, K. T. Soares, M. Z. Fortes, and B. S. M. C. Borba, "Análise da qualidade de energia no caso da geração distribuição atuando em serviços ancilares.," *Revista FSA*, vol. 19, no. 12, pp. 233–246, 2022, ISSN: 18066356.

[18] O. d. T. e. E. Ministério do Ambiente, *Decreto-lei n.° 153/2014*, https://dre.pt/dre/detalhe/decreto-lei/153-2014-58406974, January, 2023.

[19] A. e Ação Climática, *Portaria n.° 80/2020*, https://dre.pt/dre/detalhe/portaria/80-2020-130659001, January, 2023.

[20] R. Faia, T. Pinto, Z. Vale, and J. M. Corchado, "Prosumer community portfolio optimization via aggregator: The case of the iberian electricity market and portuguese retail market," *Energies*, vol. 14, no. 13, 2021, ISSN: 1996-1073. DOI: 10.3390/en14133747. [Online]. Available: https://www.mdpi.com/1996-1073/14/13/3747.

[21] P. do Conselho de Ministros, *Decreto-lei n.º 15/2022*, https://dre.pt/dre/detalhe/decreto-lei/15-2022-177634016, January, 2023.

[22] J. R. S. Fialho, A. P. S. Pinto, and A. L. C. Gomes, "Avaliação económica de um sistema de produção fotovoltaico para autoconsumo," 2018.

[23] REN, *Sintese anual 2011 - 2015 - ren*, https://mercado.ren.pt/PT/Electr/InfoMercado/PressReleases/BibInfAnual/MercadoEletricidadeSinteseAnual2011_2015.pdf, January, 2023.

[24] F. Pinto, *Regime jurídico das unidades de produção (up) distribuída*, https://www.ordemengenheiros.pt/fotos/dossier_artigo/20151120_filipepinto_7615808675660254a2675c.pdf, January, 2023.

[25] R. S. Shivalkar, H. T. Jadhav, and P. Deo, "Feasibility study for the net metering implementation in rooftop solar pv installations across reliance energy consumers," in *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, 2015, pp. 1–6. DOI: 10.1109/ICCPCT.2015.7159370.

[26] ERSE, *Regulamento n.º 266/2020*, https://dre.pt/dre/detalhe/regulamento/266-2020-130469272, February, 2023.

[27] L. Pires Klein, A. Krivoglazova, L. Matos, J. Landeck, and M. De Azevedo, "A novel peer-to-peer energy sharing business model for the portuguese energy market," *Energies*, vol. 13, no. 1, p. 125, 2019.

[28] L. M. Carvalho, "Gestão e operação de comunidades de energias renováveis com integração de baterias," M.S. thesis, Faculdade de Engenharia da Universidade do Porto, 2022, p. 31.

[29] F. M. Vieira, P. S. Moura, and A. T. de Almeida, "Energy storage system for self-consumption of photovoltaic energy in residential zero energy buildings," *Renewable Energy*, vol. 103, pp. 308–320, 2017, ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2016.11.048. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148116310321.

[30] S. Comello and S. Reichelstein, "The emergence of cost effective battery storage," *Nature Communications*, vol. 10, no. 1, p. 2038, 2019.

[31] G. Zubi, R. Dufo-López, M. Carvalho, and G. Pasaoglu, "The lithium-ion battery: State of the art and future perspectives," *Renewable and Sustainable Energy Reviews*, vol. 89, pp. 292–308, 2018, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2018.03.002. [Online]. Available: %5Curl%7Bhttps://www.sciencedirect.com/science/article/pii/S1364032118300728%7D.

[32] M. Stecca, L. R. Elizondo, T. B. Soeiro, P. Bauer, and P. Palensky, "A comprehensive review of the integration of battery energy storage systems into distribution networks," *IEEE Open Journal of the Industrial Electronics Society*, vol. 1, pp. 46–65, 2020. DOI: 10.1109/OJIES.2020.2981832.

[33] A. W. F. Wesley Cole and C. Augustine, *Cost projections for utility-scale battery storage: 2021 update*, https://www.nrel.gov/docs/fy21osti/79236.pdf, February, 2023.

[34] K. Iba, "Massive energy storage system for effective usage of renewable energy," *Global Energy Interconnection*, vol. 5, no. 3, pp. 301–308, 2022, ISSN: 2096-5117. DOI: https://doi.org/10.1016/j.gloei.2022.06.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2096511722000627.

[35] U. Mulleriyawage and W. Shen, "Optimally sizing of battery energy storage capacity by operational optimization of residential pv-battery systems: An australian household case study," *Renewable Energy*, vol. 160, pp. 852–864, 2020, ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2020.07.022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960148120310983.

[36] H. Averfalk, P. Ingvarsson, U. Persson, and S. Werner, "On the use of surplus electricity in district heating systems," in *Proceedings from the 14th International Symposium on District Heating and Cooling : September, 6-10, 2014: Stockholm, Sweden*, The work presented in this paper is a result of the research activities of the Strategic Research Centre for 4th Generation District Heating (4DH), which has received funding from The Danish Council for Strategic Research., Swedish District Heating Association, 2014, pp. 469–474, ISBN: 978-91-85775-24-8.

[37] M. C. Jesus, P. G. S. Dias, Â. R. Oliveira, and L. O. A. Junior, "Sistemas fotovoltaicos na produção rural," in *XV Encontro Mineiro de Engenharia de Produção*, vol. 1, 2019, pp. 1–9. DOI: https://doi.org/10.18407/issn.1983-0629.2019.

[38] B. A. Bhayo, H. H. Al-Kayiem, and S. I. Gilani, "Assessment of standalone solar pv-battery system for electricity generation and utilization of excess power for water pumping," *Solar Energy*, vol. 194, pp. 766–776, 2019, ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2019.11.026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X19311260.

[39] D. H. Clift and H. Suehrcke, "Control optimization of pv powered electric storage and heat pump water heaters," *Solar Energy*, vol. 226, pp. 489–500, 2021, ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2021.08.059. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0038092X21007234.

[40] K. Scott, "Introduction to Electrolysis, Electrolysers and Hydrogen Production," in *Electrochemical Methods for Hydrogen Production*, The Royal Society of Chemistry, Nov. 2019, ISBN: 978-1-78801-378-9. DOI: 10.1039/9781788016049-00001. [Online]. Available: https://doi.org/10.1039/9781788016049-00001.

[41] V. S. Tabar and V. Abbasi, "Energy management in microgrid with considering high penetration of renewable resources and surplus power generation problem," *Energy*, vol. 189, p. 116 264, 2019, ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.116264. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544219319590.

[42] P. G. S. Dias, T. Brito, L. C. G. Lopes, and J. Lima, "Smart system for monitoring and controlling energy consumption and ambient conditions," in *VII CIEEMAT 2022 – Congresso Ibero-Americano de Empreendedorismo, Energia, Ambiente e Tecnologia*, Instituto Politécnico de Bragança, 2022.

[43] E. Nodar Carro, *Análise de consumo enerxético mediante tecnoloxías iot*, http://hdl.handle.net/10198/22817, April, 2022.

[44] J. Queiroz, J. Barbosa, J. Dias, P. Leitão, and E. Oliveira, "Development of a smart electric motor testbed for internet of things and big data technologies," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 3435–3440. DOI: 10.1109/IECON.2017.8216582.

[45] J. A. Hernández, C. K. Franco, D. A. Avila, and J. A. Murillo, "Design and implementation of a management system of surplus energy generated by a distributed generation system, case study gcpvs," in *2014 IEEE 40th Photovoltaic Specialist Conference (PVSC)*, 2014, pp. 2734–2739. DOI: 10.1109/PVSC.2014.6925494.

[46] A. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar, "A smart home energy management system using iot and big data analytics approach," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 426–434, 2017. DOI: 10.1109/TCE.2017.015014.

[47] S. Z. R. Hussain, A. Osman, M. A. Moin, and J. A. Memon, "Iot enabled real-time energy monitoring and control system," in *2021 9th International Conference on Smart Grid (icSmartGrid)*, 2021, pp. 97–102. DOI: 10.1109/icSmartGrid52357.2021.9551208.

[48] K. A. Silva, *Monitor de consumo de energia elétrica com conexão à internet*, https://repositorio.ufu.br/handle/123456789/24832, April, 2022.

[49] G. C. B. R. Carvalho, *Load management in a smart house*, https://hdl.handle.net/10216/119117, April, 2022.

[50] MULIADI, M. Y. FAHREZI, I. S. ARENI, E. PALANTEI, and A. ACHMAD, "A smart home energy consumption monitoring system integrated with internet connection," in *2020 IEEE International Conference on Communication, Networks and Satellite (Comnetsat)*, 2020, pp. 75–80. DOI: 10.1109/Comnetsat50391.2020.9328960.

[51] A. Chaudhari, B. Rodrigues, and S. More, "Automated iot based system for home automation and prediction of electricity usage and comparative analysis of various electricity providers: Smartplug," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 2016, pp. 390–392. DOI: 10.1109/IC3I.2016.7917995.

[52] S. Paul and N. P. Padhy, "Real-time energy management for smart homes," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4177–4188, 2021. DOI: 10.1109/JSYST.2020.3016358.

[53] J. A. Hernández, C. K. Franco, D. A. Avila, and J. A. Murillo, "Design and implementation of a management system of surplus energy generated by a distributed generation system, case study gcpvs," in *2014 IEEE 40th Photovoltaic Specialist Conference (PVSC)*, 2014, pp. 2734–2739. DOI: 10.1109/PVSC.2014.6925494.

[54] M. Rastegar, M. Fotuhi-Firuzabad, and H. Zareipour, "Home energy management incorporating operational priority of appliances," *International Journal of Electrical Power & Energy Systems*, vol. 74, pp. 286–292, 2016, ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2015.07.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014206151500321X.

[55] M. Shakeri, N. Amin, J. Pasupuleti, *et al.*, "An autonomous home energy management system using dynamic priority strategy in conventional homes," *Energies*, vol. 13, no. 13, 2020, ISSN: 1996-1073. DOI: 10.3390/en13133312. [Online]. Available: https://www.mdpi.com/1996-1073/13/13/3312.

[56] S. Zhai, Z. Wang, X. Yan, and G. He, "Appliance flexibility analysis considering user behavior in home energy management system using smart plugs," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1391–1401, 2019. DOI: 10.1109/TIE.2018.2815949.

[57] M. R. Sunny, M. A. Kabir, I. T. Naheen, and M. T. Ahad, "Residential energy management: A machine learning perspective," in *2020 IEEE Green Technologies Conference(GreenTech)*, 2020, pp. 229–234. DOI: 10.1109/GreenTech46478.2020.9289737.

[58] R. G. Rajasekaran, S. Manikandaraj, and R. Kamaleshwar, "Implementation of machine learning algorithm for predicting user behavior and smart energy management," in *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, 2017, pp. 24–30. DOI: 10.1109/ICDMAI.2017.8073480.

[59] M. R. A. Refaai, S. N. R. Vonteddu, P. K. Nunna, P. S. Kumar, C. Anbu, and M. Markos, "Energy management prediction in hybrid pv-battery systems using deep learning architecture," *International Journal of Photoenergy*, vol. 2022, pp. 1–7, 2022. DOI: https://doi.org/10.1155/2022/6844853.

[60] S. Asadi, S. S. Amiri, and M. Mottahedi, "On the development of multi-linear regression analysis to assess energy consumption in the early stages of building design," *Energy and Buildings*, vol. 85, pp. 246–255, 2014, ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2014.07.096. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778814007154.

[61] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things," in *2017 Internet Technologies and Applications (ITA)*, 2017, pp. 143–148. DOI: 10.1109/ITECHA.2017.8101926.

[62] E. Systems, https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, November, 2022.

[63] U. Pujari, D. Patil, D. Bahadure, M. Asnodkar, *et al.*, "Internet of things based integrated smart home automation system," in *2nd International Conference on Communication & Information Processing (ICCIP)*, 2020.

[64] A. Cornel - Cristian, T. Gabriel, M. Arhip-Calin, and A. Zamfirescu, "Smart home automation with mqtt," in *2019 54th International Universities Power Engineering Conference (UPEC)*, 2019, pp. 1–5. DOI: 10.1109/UPEC.2019.8893617.

[65] J. C. Hastings and D. M. Laverty, "Modernizing wide-area grid communications for distributed energy resource applications using mqtt publish-subscribe protocol," in *2017 IEEE Power & Energy Society General Meeting*, 2017, pp. 1–5. DOI: 10.1109/PESGM.2017.8274486.

[66] K. Ferencz and J. Domokos, "Using node-red platform in an industrial environment," Feb. 2020.

[67] P. Macheso, T. D. Manda, S. Chisale, N. Dzupire, J. Mlatho, and D. Mukanyiligira, "Design of esp8266 smart home using mqtt and node-red," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 502–505. DOI: 10.1109/ICAIS50930.2021.9396027.

[68] S. N. Z. Naqvi, S. Yfantidou, and E. Zimányi, "Time series databases and influxdb," *Studienarbeit, Université Libre de Bruxelles*, vol. 12, 2017.

[69] I. Petre, R. Boncea, C. Z. Radulescu, A. Zamfiroiu, and I. Sandu, "A time-series database analysis based on a multi-attribute maturity model," *Studies in Informatics and Control*, vol. 28, no. 2, pp. 177–188, 2019.

[70] M. Cencetti, "Aam national campaign tech talk: Grafana," 2022.

[71] M. Yang and M. Huang, "An microservices-based openstack monitoring tool," in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 2019, pp. 706–709. DOI: 10.1109/ICSESS47205.2019.9040740.

[72] B. K. Akhmetzhanov, O. A. Gazizuly, Z. Nurlan, and N. Zhakiyev, "Integration of a video surveillance system into a smart home using the home assistant platform," in *2022 International Conference on Smart Information Systems and Technologies (SIST)*, 2022, pp. 1–5. DOI: 10.1109/SIST54437.2022.9945718.

[73] A. Khusnutdinov, D. Usachev, M. Mazzara, A. Khan, and I. Panchenko, "Open source platform digital personal assistant," in *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2018, pp. 45–50. DOI: 10.1109/WAINA.2018.00062.

[74] P. G. S. Dias, T. Brito, L. C. G. Lopes, and J. Lima, "Smart system for monitoring and controlling energy consumption by residence production and load," in *Symposium of Applied Science for Young Researchers*, Viana do Castelo, 2022.

[75] S. Sah, "Machine learning: A review of learning types," 2020.

[76] S. Rong and Z. Bao-Wen, "The research of regression model in machine learning field," in *MATEC Web of Conferences*, EDP Sciences, vol. 176, 2018, p. 01 033.

[77] LEM, *Current transducer lts 25-np*, https://www.lem.com/sites/default/files/products_datasheets/lts_25-np.pdf, February, 2023.

[78] O. P. S. Data, *Data package household data*, https://data.open-power-system-data.org/household_data/2020-04-15, February, 2023.

[79] M. A. Holmstrom and D. Liu, "Machine learning applied to weather forecasting," 2016.

[80] A. McQuistan, *Using machine learning to predict the weather: Part 1*, https://stackabuse.com/using-machine-learning-to-predict-the-weather-part-1/, February, 2023.

# Appendix A

# Original Project Proposal


**Curso de Mestrado em Engenharia Industrial - Eletrotécnica**

Ano lectivo de 2022/2023

# Sistema inteligente de monitoramento e controle de consumo de energia por cargas residenciais

**Orientador: José Luís Sousa de Magalhães Lima**

**Co-orientador: Luis Cláudio Gambôa Lopes / Thadeu Vinícius de Brito**

## 1 Objectivo

O objetivo desse trabalho é desenvolver um sistema inteligente de monitoramento e controle do consumo de energia por cargas residenciais conectadas a smart plugs com base na energia excedente gerada. Baseado nas informações de energia disponível, o direcionamento desse excesso poderá ser feito manualmente ou de forma automática, por meio de um sistema inteligente que utiliza Machine Learning.

## 2 Detalhes

O usuário de energia elétrica residencial será capaz de visualizar o consumo de energia e o excedente gerado em uma interface além de controlar o consumo de energia de determinadas cargas, baseado na energia em excesso gerada. Pretende-se direcionar o excedente de energia para dispositivos elétricos fundamentando-se na energia disponível e no histórico de consumo do usuário. Ao invés de enviar a energia em excesso para um banco de baterias ou para a rede elétrica, o usuário pode utilizá-la em aplicações diárias. Esse direcionamento poderá ser feito de forma manual para cargas específicas com base na escolha do consumidor ou de forma inteligente, utilizando método de previsão baseado em Machine Learning.

## 3 Metodologia de trabalho

O desenvolvimento do projeto necessita da utilização de ferramentas de processamento, armazenamento e monitoramento de dados. Dessa forma, serão utilizadas as plataformas Node-RED, InfluxDB, Grafana e Home Assistant. Por meio desta última plataforma também será possível que o usuário controle o consumo de energia de determinadas cargas remotamente. Para que o usuário realize o monitoramento do excedente de energia gerado, especialmente, será desenvolvido um protótipo de hardware capaz de realizar a medição da potência excedente e enviar essa informação ao sistema de monitoramento. Tendo como base o histórico de geração excedente, o sistema inteligente utilizará um método de Machine Learning, através do qual será possível prever a geração de energia excendente.

| |
|---|
| **Dimensão da equipa: 4 pessoas**<br>**Recursos necessários: Recursos de hardware disponibilizados nos laboratórios do Centro de Investigação em Digitalização e Robótica Inteligente (CeDRI) e recursos de software.** |

# Appendix B

# C programming code for measurement and communication with MQTT broker

```
1  #include <ACS712.h>
2  #include <WiFi.h>
3  #include <PubSubClient.h>
4  #include "esp_adc_cal.h"
5  #include <driver/adc.h>
6
7
8  #define TOPICO_SUBSCRIBE_POTENCIA_CARGA
       "topico_potencia_carga"
9  #define TOPICO_PUBLISH_TAXA   "GRID_INFO"
10 #define TOPICO_PUBLISH_POTENCIA_EXCEDENTE
       "topico_potencia_excedente"
11 #define TOPICO_PUBLISH_TENSAO "topico_tensao"
12 #define TOPICO_PUBLISH_CORRENTE "topico_corrente"
```

```
13 #define ID_MQTT  "ESP_PLUG_IPB"
14
15 #define TENSAO 33
16 #define CORRENTE 32
17 #define LED 12
18
19
20 const char* SSID = "";
21 const char* PASSWORD = "";
22
23 const char* BROKER_MQTT = "broker.mqtt-dashboard.com";
24 int BROKER_PORT = 1883;
25
26 WiFiClient espClient;
27 PubSubClient MQTT(espClient);
28
29 int Vrede = 230;
30 int Vsensor = 1.87;
31
32 int sensorValue_aux = 0;
33 int sensorValue_aux2 = 0;
34 float voltage = 0;
35 float valorSensor = 0;
36 float valorCorrente = 0;
37 float voltsporUnidade = 0.0008056641;// 3.3/4096
38
39 float ajuste_sensor = 0;
40 float ajuste_tensao = 0;
41
```

```
42 //  Para ACS712 de 30 Amperes, usar 0.066
43 float sensibilidade = 0.066 * (3.3 / 5.0);
44
45 //float sensibilidade = 0.025;
46
47 //ACS712 sensor(ACS712_30A, CORRENTE);
48 ACS712  ACS(32, 5, 4095, 66);
49
50 float message = 0;
51 float Pexcedente = 0;
52 float Potencia[1000] = {0};
53 float Pcarga = 0;
54 float taxa = 0;
55
56
57
58 void initWiFi(void);
59 void initMQTT(void);
60 void mqtt_callback(char* topic, byte* payload, unsigned int
      length);
61 void reconnectMQTT(void);
62 void reconnectWiFi(void);
63 void VerificaConexoesWiFIEMQTT(void);
64
65 void initWiFi(void)
66 {
67   delay(10);
68   Serial.println("------Conexao WI-FI------");
69   Serial.print("Conectando-se na rede: ");
```

```
70    Serial.println(SSID);
71    Serial.println("Aguarde");
72
73    reconnectWiFi();
74  }
75
76  void initMQTT(void)
77  {
78    MQTT.setServer(BROKER_MQTT, BROKER_PORT);
79    MQTT.setCallback(mqtt_callback);
80  }
81
82  void mqtt_callback(char* topic, byte* payload, unsigned int
       length)
83  {
84    String msg;
85
86    for (int i = 0; i < length; i++)
87    {
88      char c = (char)payload[i];
89      msg += c;
90    }
91
92    Serial.print("Chegou a seguinte string via MQTT: ");
93    Serial.println(msg);
94
95    message = msg.toFloat();
96
97    if (msg.equals("on"))
```

```
98    {
99      Serial.print("LED aceso mediante comando MQTT");
100     digitalWrite(LED, HIGH);
101   }
102
103   if (msg.equals("off"))
104   {
105     Serial.print("LED apagado mediante comando MQTT");
106     digitalWrite(LED, LOW);
107   }
108 }
109
110
111 void reconnectMQTT(void)
112 {
113   while (!MQTT.connected())
114   {
115     Serial.print("* Tentando se conectar ao Broker MQTT: ");
116     Serial.println(BROKER_MQTT);
117     if (MQTT.connect(ID_MQTT))
118     {
119       Serial.println("Conectado com sucesso ao broker MQTT!");
120       MQTT.subscribe(TOPICO_SUBSCRIBE_POTENCIA_CARGA);
121     }
122     else
123     {
124       Serial.println("Falha ao reconectar no broker.");
125       Serial.println("Havera nova tentativa de conexao em
            2s");
```

```
126         delay(2000);
127       }
128     }
129   }
130
131
132   void VerificaConexoesWiFIEMQTT(void)
133   {
134     if (!MQTT.connected())
135       reconnectMQTT();
136
137     reconnectWiFi();
138   }
139
140
141   void reconnectWiFi(void)
142   {
143     if (WiFi.status() == WL_CONNECTED)
144       return;
145
146     WiFi.begin(SSID, PASSWORD); // Conecta na rede WI-FI
147
148     while (WiFi.status() != WL_CONNECTED)
149     {
150       delay(100);
151       Serial.print(".");
152     }
153
154     Serial.println();
```

```
155     Serial.print("Conectado com sucesso na rede ");
156     Serial.print(SSID);
157     Serial.println("IP obtido: ");
158     Serial.println(WiFi.localIP());
159  }
160
161
162  void setup() {
163     pinMode(TENSAO, INPUT);
164     pinMode(CORRENTE, INPUT);
165     Serial.begin(115200);
166
167     initWiFi();
168
169     initMQTT();
170
171     for (int i = 0; i < 1000; i++) {
172       ajuste_sensor += analogRead(CORRENTE);
173     }
174     ajuste_sensor = ajuste_sensor / 1000;
175     Serial.println(ajuste_sensor);
176
177
178     for (int i = 0; i < 1000; i++) {
179       ajuste_tensao += analogRead(TENSAO);
180     }
181     ajuste_tensao = ajuste_tensao / 1000;
182     Serial.println(ajuste_tensao);
183   }
```

```
184
185
186 void loop() {
187   float amostras_tensao[1000];
188   float tensao[1000];
189   float tensaoRMS;
190   float soma_tensao = 0;
191   float corrente[1000] = {0};
192   float amostras_corrente[1000];
193   float soma_corrente = 0;
194   float saida_sensor[1000];
195   float soma_potencia = 0;
196   float Ppos = 0;
197   float Pneg = 0;
198
199   char porcentagem [10] = {0};
200   char excedente [10] = {0};
201   char voltageMQTT [12] = {0};
202   char currentMQTT [10] = {0};
203
204 //CALCULO DA CORRENTE, TENSAO E POTENCIA
205
206  for (int i = 1000; i > 0; i--) {
207
208      amostras_tensao[i] = analogRead(TENSAO);
209      tensao[i] = ((amostras_tensao[i] - ajuste_tensao) *
           voltsporUnidade) * Vrede/Vsensor;// * (Vrede /
           (ajuste_tensao * voltsporUnidade));
210
```

```
211       sensorValue_aux = (analogRead(CORRENTE) - ajuste_sensor);
212       sensorValue_aux2 = (amostras_tensao[i] - ajuste_tensao);
213
214       //corrente[i] = ((sensorValue_aux * voltsporUnidade) /
              sensibilidade) / 3.0;
215       Serial.println(corrente[i]);
216       corrente[i] = ((sensorValue_aux * voltsporUnidade) /
              sensibilidade);
217       Potencia[i] = tensao[i] * corrente[i];
218
219       valorSensor += pow(sensorValue_aux, 2);
220       voltage += pow(sensorValue_aux2, 2);
221
222     if(Potencia[i] > 0){
223       Ppos++;
224     }
225     else{
226       Pneg++;
227     }
228       soma_potencia += pow(Potencia[i], 2);
229   }
230
231   // Calculo da corrente RMS
232     valorSensor = (sqrt(valorSensor / 1000)) *
            voltsporUnidade;
233     valorCorrente = (valorSensor / sensibilidade);
234     //valorCorrente = valorCorrente / 3.0; //para ACS712 30A
            nao usa essa relacao
235
```

```
236       if(valorCorrente <= 0.65){ //para retirada de ruido
237        valorCorrente = 0;
238       }
239
240       valorSensor = 0;
241
242       Serial.print("Corrente : ");
243       // Corrente
244       Serial.print(valorCorrente, 3);
245       Serial.println(" A ");
246
247
248    // Calculo da tensao RMS
249    voltage = (sqrt(voltage / 1000));
250    voltage = (voltage  * voltsporUnidade) * Vrede/Vsensor;
251
252     Serial.print("Tensao : ");
253      // Tensao
254     Serial.print(voltage, 3);
255     Serial.println(" V ");
256
257
258    if(Ppos > Pneg){
259       Pexcedente = sqrt(soma_potencia / 1000);
260       Serial.print(" Potencia Excedente: ");
261       Serial.print(Pexcedente);
262       Serial.println(" Watts ");
263    }
264    else{
```

```
265       Pexcedente = 0;
266       Serial.println(" Nao ha potencia excedente.");
267     }
268
269     sprintf(excedente, "%.f", Pexcedente);
270
271     VerificaConexoesWiFIEMQTT();
272
273     Pcarga = message;
274
275     if (Pexcedente == 0.00) {
276       taxa = 0;
277     }
278     else if (Pcarga > Pexcedente) {
279       taxa = (Pexcedente / Pcarga) * 100.0;
280     }
281
282     else {
283       taxa = 100.0;
284     }
285
286     sprintf(porcentagem, "%.f", taxa);
287     Serial.print("Porcentagem: ");
288     Serial.print(porcentagem);
289     Serial.println("%");
290
291     sprintf(voltageMQTT, "%f", voltage);
292     Serial.print("Tensao: ");
293     Serial.print(voltageMQTT);
```

```
294     Serial.println("V");
295
296     sprintf(currentMQTT, "%3.2f", valorCorrente);
297     Serial.print("Corrente: ");
298     Serial.print(currentMQTT);
299     Serial.println("A");
300
301
302     MQTT.publish(TOPICO_PUBLISH_POTENCIA_EXCEDENTE, excedente);
303     MQTT.publish(TOPICO_PUBLISH_TAXA, porcentagem);
304     MQTT.publish(TOPICO_PUBLISH_TENSAO, voltageMQTT);
305     MQTT.publish(TOPICO_PUBLISH_CORRENTE, currentMQTT);
306
307
308     Serial.println("Mensagem enviada ao broker");
309
310     MQTT.loop();
311
312     delay(2000);
313
314 }
```

# Appendix C

# Data for the first analysis of Linear Regression

| time | mean_surplus | X1 | X2 | X3 | X4 | X5 | X6 |
|---|---|---|---|---|---|---|---|
| 2016-05-14T00:00:00Z | -0.007 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2016-05-14T00:15:00Z | -0.007 | -0.007 | 0 | 0 | 0 | 0 | 0 |
| 2016-05-14T00:30:00Z | -0.007 | -0.007 | -0.007 | 0 | 0 | 0 | 0 |
| 2016-05-14T00:45:00Z | -0.007 | -0.007 | -0.007 | -0.007 | 0 | 0 | 0 |
| 2016-05-14T01:00:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | 0 | 0 |
| 2016-05-14T01:15:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | 0 |
| 2016-05-14T01:30:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T01:45:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T02:00:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T02:15:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T02:30:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T02:45:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T03:00:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T03:15:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2016-05-14T03:30:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T03:45:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T04:00:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T04:15:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T04:30:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T04:45:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T05:00:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T05:15:00Z | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T05:30:00Z | -0.008 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T05:45:00Z | 0.018 | -0.008 | -0.007 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T06:00:00Z | 0.044 | 0.018 | -0.008 | -0.007 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T06:15:00Z | 0.03 | 0.044 | 0.018 | -0.008 | -0.007 | -0.007 | -0.007 |
| 2016-05-14T06:30:00Z | 0.005 | 0.03 | 0.044 | 0.018 | -0.008 | -0.007 | -0.007 |
| 2016-05-14T06:45:00Z | 0.102 | 0.005 | 0.03 | 0.044 | 0.018 | -0.008 | -0.007 |
| 2016-05-14T07:00:00Z | 0.172 | 0.102 | 0.005 | 0.03 | 0.044 | 0.018 | -0.008 |
| 2016-05-14T07:15:00Z | 0.149 | 0.172 | 0.102 | 0.005 | 0.03 | 0.044 | 0.018 |
| 2016-05-14T07:30:00Z | 0.234 | 0.149 | 0.172 | 0.102 | 0.005 | 0.03 | 0.044 |
| 2016-05-14T07:45:00Z | 0.3 | 0.234 | 0.149 | 0.172 | 0.102 | 0.005 | 0.03 |
| 2016-05-14T08:00:00Z | 0.482 | 0.3 | 0.234 | 0.149 | 0.172 | 0.102 | 0.005 |
| 2016-05-14T08:15:00Z | 0.56 | 0.482 | 0.3 | 0.234 | 0.149 | 0.172 | 0.102 |
| 2016-05-14T08:30:00Z | 0.484 | 0.56 | 0.482 | 0.3 | 0.234 | 0.149 | 0.172 |
| 2016-05-14T08:45:00Z | 0.547 | 0.484 | 0.56 | 0.482 | 0.3 | 0.234 | 0.149 |
| 2016-05-14T09:00:00Z | 0.699 | 0.547 | 0.484 | 0.56 | 0.482 | 0.3 | 0.234 |
| 2016-05-14T09:15:00Z | 0.637 | 0.699 | 0.547 | 0.484 | 0.56 | 0.482 | 0.3 |
| 2016-05-14T09:30:00Z | 0.767 | 0.637 | 0.699 | 0.547 | 0.484 | 0.56 | 0.482 |
| 2016-05-14T09:45:00Z | 0.783 | 0.767 | 0.637 | 0.699 | 0.547 | 0.484 | 0.56 |
| 2016-05-14T10:00:00Z | 1.165 | 0.783 | 0.767 | 0.637 | 0.699 | 0.547 | 0.484 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2016-05-14T10:15:00Z | 0.906 | 1.165 | 0.783 | 0.767 | 0.637 | 0.699 | 0.547 |
| 2016-05-14T10:30:00Z | 0.773 | 0.906 | 1.165 | 0.783 | 0.767 | 0.637 | 0.699 |
| 2016-05-14T10:45:00Z | 0.997 | 0.773 | 0.906 | 1.165 | 0.783 | 0.767 | 0.637 |
| 2016-05-14T11:00:00Z | 0.529 | 0.997 | 0.773 | 0.906 | 1.165 | 0.783 | 0.767 |
| 2016-05-14T11:15:00Z | 0.482 | 0.529 | 0.997 | 0.773 | 0.906 | 1.165 | 0.783 |
| 2016-05-14T11:30:00Z | 0.039 | 0.482 | 0.529 | 0.997 | 0.773 | 0.906 | 1.165 |
| 2016-05-14T11:45:00Z | 0.003 | 0.039 | 0.482 | 0.529 | 0.997 | 0.773 | 0.906 |
| 2016-05-14T12:00:00Z | 0.042 | 0.003 | 0.039 | 0.482 | 0.529 | 0.997 | 0.773 |
| 2016-05-14T12:15:00Z | 0.046 | 0.042 | 0.003 | 0.039 | 0.482 | 0.529 | 0.997 |
| 2016-05-14T12:30:00Z | 0.104 | 0.046 | 0.042 | 0.003 | 0.039 | 0.482 | 0.529 |
| 2016-05-14T12:45:00Z | 0.183 | 0.104 | 0.046 | 0.042 | 0.003 | 0.039 | 0.482 |
| 2016-05-14T13:00:00Z | 0.123 | 0.183 | 0.104 | 0.046 | 0.042 | 0.003 | 0.039 |
| 2016-05-14T13:15:00Z | 0.247 | 0.123 | 0.183 | 0.104 | 0.046 | 0.042 | 0.003 |
| 2016-05-14T13:30:00Z | 0.342 | 0.247 | 0.123 | 0.183 | 0.104 | 0.046 | 0.042 |
| 2016-05-14T13:45:00Z | 0.313 | 0.342 | 0.247 | 0.123 | 0.183 | 0.104 | 0.046 |
| 2016-05-14T14:00:00Z | 0.227 | 0.313 | 0.342 | 0.247 | 0.123 | 0.183 | 0.104 |
| 2016-05-14T14:15:00Z | 0.313 | 0.227 | 0.313 | 0.342 | 0.247 | 0.123 | 0.183 |
| 2016-05-14T14:30:00Z | 0.52 | 0.313 | 0.227 | 0.313 | 0.342 | 0.247 | 0.123 |
| 2016-05-14T14:45:00Z | 0.2 | 0.52 | 0.313 | 0.227 | 0.313 | 0.342 | 0.247 |
| 2016-05-14T15:00:00Z | 0.032 | 0.2 | 0.52 | 0.313 | 0.227 | 0.313 | 0.342 |
| 2016-05-14T15:15:00Z | 0.096 | 0.032 | 0.2 | 0.52 | 0.313 | 0.227 | 0.313 |
| 2016-05-14T15:30:00Z | 0.368 | 0.096 | 0.032 | 0.2 | 0.52 | 0.313 | 0.227 |
| 2016-05-14T15:45:00Z | 1.187 | 0.368 | 0.096 | 0.032 | 0.2 | 0.52 | 0.313 |
| 2016-05-14T16:00:00Z | 0.464 | 1.187 | 0.368 | 0.096 | 0.032 | 0.2 | 0.52 |
| 2016-05-14T16:15:00Z | 0.128 | 0.464 | 1.187 | 0.368 | 0.096 | 0.032 | 0.2 |
| 2016-05-14T16:30:00Z | 0.128 | 0.128 | 0.464 | 1.187 | 0.368 | 0.096 | 0.032 |
| 2016-05-14T16:45:00Z | -0.008 | 0.128 | 0.128 | 0.464 | 1.187 | 0.368 | 0.096 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2016-05-14T17:00:00Z | -0.038 | -0.008 | 0.128 | 0.128 | 0.464 | 1.187 | 0.368 |
| 2016-05-14T17:15:00Z | -0.029 | -0.038 | -0.008 | 0.128 | 0.128 | 0.464 | 1.187 |
| 2016-05-14T17:30:00Z | 0.043 | -0.029 | -0.038 | -0.008 | 0.128 | 0.128 | 0.464 |
| 2016-05-14T17:45:00Z | 0.013 | 0.043 | -0.029 | -0.038 | -0.008 | 0.128 | 0.128 |
| 2016-05-14T18:00:00Z | -0.008 | 0.013 | 0.043 | -0.029 | -0.038 | -0.008 | 0.128 |
| 2016-05-14T18:15:00Z | -0.007 | -0.008 | 0.013 | 0.043 | -0.029 | -0.038 | -0.008 |
| 2016-05-14T18:30:00Z | -0.008 | -0.007 | -0.008 | 0.013 | 0.043 | -0.029 | -0.038 |
| 2016-05-14T18:45:00Z | -0.007 | -0.008 | -0.007 | -0.008 | 0.013 | 0.043 | -0.029 |
| 2016-05-14T19:00:00Z | -0.008 | -0.007 | -0.008 | -0.007 | -0.008 | 0.013 | 0.043 |
| 2016-05-14T19:15:00Z | -0.008 | -0.008 | -0.007 | -0.008 | -0.007 | -0.008 | 0.013 |
| 2016-05-14T19:30:00Z | -0.008 | -0.008 | -0.008 | -0.007 | -0.008 | -0.007 | -0.008 |
| 2016-05-14T19:45:00Z | -0.007 | -0.008 | -0.008 | -0.008 | -0.007 | -0.008 | -0.007 |
| 2016-05-14T20:00:00Z | -0.007 | -0.007 | -0.008 | -0.008 | -0.008 | -0.007 | -0.008 |
| 2016-05-14T20:15:00Z | -0.009 | -0.007 | -0.007 | -0.008 | -0.008 | -0.008 | -0.007 |
| 2016-05-14T20:30:00Z | -0.007 | -0.009 | -0.007 | -0.007 | -0.008 | -0.008 | -0.008 |
| 2016-05-14T20:45:00Z | -0.007 | -0.007 | -0.009 | -0.007 | -0.007 | -0.008 | -0.008 |
| 2016-05-14T21:00:00Z | -0.008 | -0.007 | -0.007 | -0.009 | -0.007 | -0.007 | -0.008 |
| 2016-05-14T21:15:00Z | -0.008 | -0.008 | -0.007 | -0.007 | -0.009 | -0.007 | -0.007 |
| 2016-05-14T21:30:00Z | -0.008 | -0.008 | -0.008 | -0.007 | -0.007 | -0.009 | -0.007 |
| 2016-05-14T21:45:00Z | -0.007 | -0.008 | -0.008 | -0.008 | -0.007 | -0.007 | -0.009 |
| 2016-05-14T22:00:00Z | -0.008 | -0.007 | -0.008 | -0.008 | -0.008 | -0.007 | -0.007 |
| 2016-05-14T22:15:00Z | -0.008 | -0.008 | -0.007 | -0.008 | -0.008 | -0.008 | -0.007 |
| 2016-05-14T22:30:00Z | -0.008 | -0.008 | -0.008 | -0.007 | -0.008 | -0.008 | -0.008 |
| 2016-05-14T22:45:00Z | -0.008 | -0.008 | -0.008 | -0.008 | -0.007 | -0.008 | -0.008 |
| 2016-05-14T23:00:00Z | -0.007 | -0.008 | -0.008 | -0.008 | -0.008 | -0.007 | -0.008 |
| 2016-05-14T23:15:00Z | -0.008 | -0.007 | -0.008 | -0.008 | -0.008 | -0.008 | -0.007 |
| 2016-05-14T23:30:00Z | -0.007 | -0.008 | -0.007 | -0.008 | -0.008 | -0.008 | -0.008 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2016-05-14T23:45:00Z | -0.009 | -0.007 | -0.008 | -0.007 | -0.008 | -0.008 | -0.008 |

# Appendix D

# Code in Python used for the analysis of the Linear Regression models

```
1  import pandas as pd
2  import numpy as np
3  import statsmodels.api as sm
4  from sklearn.linear_model import LinearRegression
5  from sklearn.model_selection import train_test_split
6  from sklearn.metrics import explained_variance_score, \
7      mean_absolute_error, \
8      median_absolute_error
9
10 import matplotlib.pyplot as plt
11 from datetime import datetime
12 from influxdb import InfluxDBClient
13
14 ############### Convert excel to line protocol ###############
15
16 #convert sample data to line protocol
17 dataset = pd.read_excel('./Consumption_Generation_Data.xlsx')
18
19 dataset['utc_timestamp'] = pd.to_datetime(dataset['utc_timestamp'])
20 print(dataset)
```

```
21
22 data = dataset['utc_timestamp']
23 for i in range(len(dataset)):
24     dataset['timestamp'] = dataset['utc_timestamp'].astype("int64")
25
26 dataset['timestamp'].head()
27
28 #Rounding the surplus column values
29 dataset['surplus'] = dataset['surplus'].round(3)
30 print(dataset['surplus'])
31
32 lines = [ "Energy"
33          + ",type=kWh"
34          + " "
35          + "circulation_pump=" + str(dataset['circulation_pump '][d]) + ","
36          + "dishwasher=" + str(dataset["dishwasher"][d]) + ","
37          + "freezer=" + str(dataset["freezer "][d]) + ","
38          + "grid_export=" + str(dataset["grid_export"][d]) + ","
39          + "grid_import=" + str(dataset["grid_import "][d]) + ","
40          + "pv=" + str(dataset["pv"][d]) + ","
41          + "washing_machine=" + str(dataset["washing_machine"][d]) + ","
42          + "consumption=" + str(dataset["consumption"][d]) + ","
43          + "surplus=" + str(dataset["surplus"][d])
44          + " " + str(dataset['timestamp'][d]) for d in range(len(dataset))]
45
46 thefile = open('Consumption_Generation_Data.txt', 'w')
47 for item in lines:
48     thefile.write("%s\n" % item)
49
50 host = ''
51 port =
52 user = ''
53 password = ''
54 dbname = 'Potencia_Excedente'
55
```

```
protocol = 'line'

client = InfluxDBClient(host, port, user, password, dbname)

with open('./Consumption_Generation_Data.txt', 'r') as file:
    data = file.read()

print(data)

print("Write DataFrame")
client.write_points(data, database=dbname, protocol=protocol)

################ Query InfluxDB data ################
query = 'SELECT mean("surplus") AS "mean_surplus" FROM
    "Potencia_Excedente"."autogen"."Energy" WHERE "time" >=
    \'2016-05-14T00:00:00Z\' AND "time" <= \'2016-05-14T12:00:00Z\' GROUP
    BY time(15m) FILL(null)'

print("Querying data: " + query)
result = client.query(query)

df = pd.DataFrame(columns=['time', 'mean_surplus']) #Creating a dataframe
    to insert the consulted data

print('Point[surplus]')
for point in result.get_points():
    print(point)
print(result)

for d in result:
    df = df.append(d, ignore_index=True)

features = list(df.columns)

def derive_nth_day_feature(df, feat, n):
```

```
87      rows = df.shape[0]
88      nth_prior_measurements = [None] * n + [df[feat][i - n] for i in
            range(n, rows)]
89      col_name = "{}{}".format("X", n)
90      df[col_name] = nth_prior_measurements
91
92
93  for feature in features:
94      if feature != 'time':
95          for N in range(1, 8):
96              derive_nth_day_feature(df, feature, N)
97
98  df.fillna(0, inplace=True)
99  df.head()
100
101 with pd.option_context('display.max_rows', None, 'display.max_columns',
        None):
102     print(df)
103
104 query2 = 'SELECT mean("surplus") AS "mean_surplus" FROM
        "Potencia_Excedente"."autogen"."Energy" WHERE "time" >=
        \'2016-05-15T00:00:00Z\' AND "time" <= \'2016-06-14T23:45:00Z\' GROUP
        BY time(15m) FILL(null)'
105
106 print("Querying data: " + query2)
107 result = client.query(query2)
108
109 for point in result.get_points():
110     print(point)
111
112 df_day2 = pd.DataFrame(result.get_points())
113
114 df_day2.fillna(0, inplace=True)
115
116 print(df_day2)
```

```
117
118 with pd.option_context('display.max_rows', None, 'display.max_columns',
        None):
119     print(df_day2)
120
121 ################ Linear Regression with InfluxDB data ################
122
123 #predictors = ['X1', 'X2', 'X3', 'X4', 'X5', 'X6']
124 #predictors = ['X1', 'X2', 'X3', 'X4', 'X5']
125 #predictors = ['X1', 'X2', 'X3', 'X4']
126 #predictors = ['X1', 'X2', 'X3']
127 #predictors = ['X1', 'X2']
128 predictors = ['X1']
129 Y = df['mean_surplus']
130 X = df[predictors]
131
132 plt.rcParams['figure.figsize'] = [22, 16]
133
134 fig, axes = plt.subplots(nrows=2, ncols=3, sharey=True)
135 arr = np.array(predictors).reshape(2, 3)
136 for row, col_arr in enumerate(arr):
137     for col, feature in enumerate(col_arr):
138         axes[row, col].scatter(df[feature], df['mean_surplus'])
139         if col == 0:
140             axes[row, col].set(xlabel=feature, ylabel='mean_surplus')
141         else:
142             axes[row, col].set(xlabel=feature)
143 plt.show()
144
145 #Obtaining coefficients a and b from the OLS function of the StatsModel
        library
146 X = sm.add_constant(X)
147 modelo = sm.OLS(Y, X).fit()
148 modelo.summary()
```

```
149 X = X.drop('const', axis = 1) #Take the value of column X and a column of
        constants and eliminate the entire column of constants
150 Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y, test_size=0.2,
        random_state = 12) # 20% of the data is test, random_state define the
        random selection of data
151 print(Xtrain.shape, Xtest.shape, Ytrain.shape, Ytest.shape) #Show the
        number of rows and columns for these variables
152
153 #Application of Linear Regression
154 mod = LinearRegression()
155 mod.fit(Xtrain, Ytrain) #Fit the model to the data. Estimation of
        coefficients. It finds the values of a and b based on the values of X
        and Y
156 p = mod.predict(Xtest)
157 p
158 accuracy = mod.score(Xtest, Ytest) #referente ao modelo
159 print("Accuracy = {:0.2f}".format(accuracy))
160
161 coef_angular = mod.coef_[0]
162 coef_linear = mod.intercept_
163
164 print("Angular Coefficient = ")
165 print(coef_angular)
166 print("Linear Coefficient = ")
167 print(coef_linear)
168
169 real_surplus = df_day2['mean_surplus']
170
171 Z = df_day2['mean_surplus']
172
173 for i in range(len(df_day2)):
174     df_day2.loc[i, 'Predicted_data'] =
            coef_angular*(Z.values[i])+coef_linear
175     if (df_day2.loc[i, "Predicted_data"] < 0):
176         df_day2.loc[i, "Predicted_data"] = 0
```

```
177     print(df_day2.loc[i, "Predicted_data"])
178     df_day2.loc[i, "Error"] = df_day2.loc[i,'mean_surplus'] -
            df_day2.loc[i,'Predicted_data']
179
180 with pd.option_context('display.max_rows', None, 'display.max_columns',
        None):
181     print(df_day2["Predicted_data"])
182
183 with pd.option_context('display.max_rows', None, 'display.max_columns',
        None):
184     print(df_day2)
185
186 mean_absolute_error = mean_absolute_error(real_surplus,
        df_day2["Predicted_data"])
187 print(mean_absolute_error)
188 median_absolute_error = median_absolute_error(real_surplus,
        df_day2["Predicted_data"])
189 print(median_absolute_error)
190 explained_variance_score = explained_variance_score(real_surplus,
        df_day2["Predicted_data"])
191 print(explained_variance_score)
192
193 erro = df_day2.Error.to_numpy()
194 var_erro = np.var(erro, dtype=np.float64)
195
196 print(var_erro)
197
198 reta = coef_angular*X+coef_linear
199 plt.scatter(X, Y)
200 plt.plot(X, reta, label = 'Linear Fit', color='red')
201 plt.xlabel('Energy 1')
202 plt.ylabel('Energy')
203 plt.legend()
```

# Appendix E

# Prediction results with models created from hours

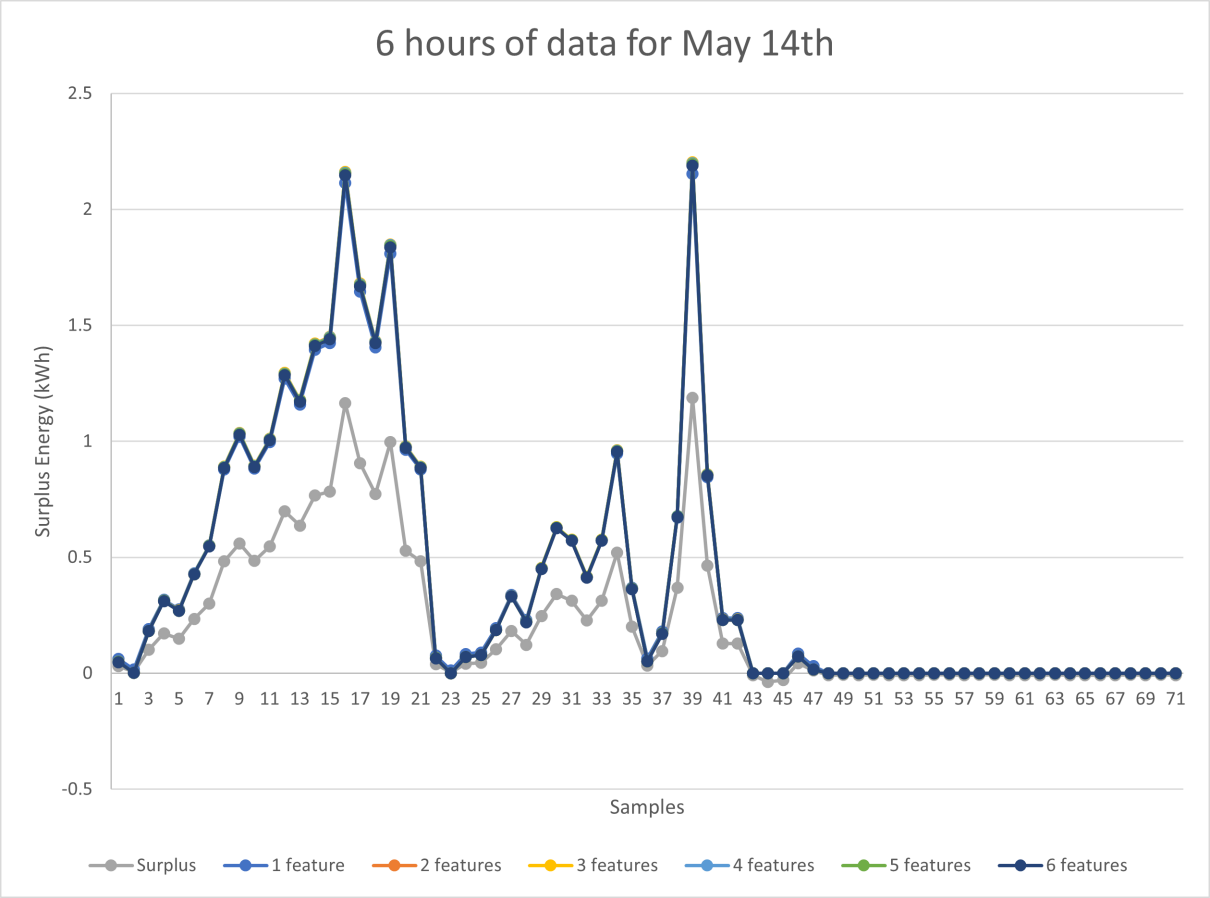


Figure E.1: Graphical representation of forecasts using 6 hours of day 14 to forecast the rest of the day. Due to the proximity of the values obtained with the models that use 1 to 6 features as X predictor values, the graphical visualization of all the results was impaired.

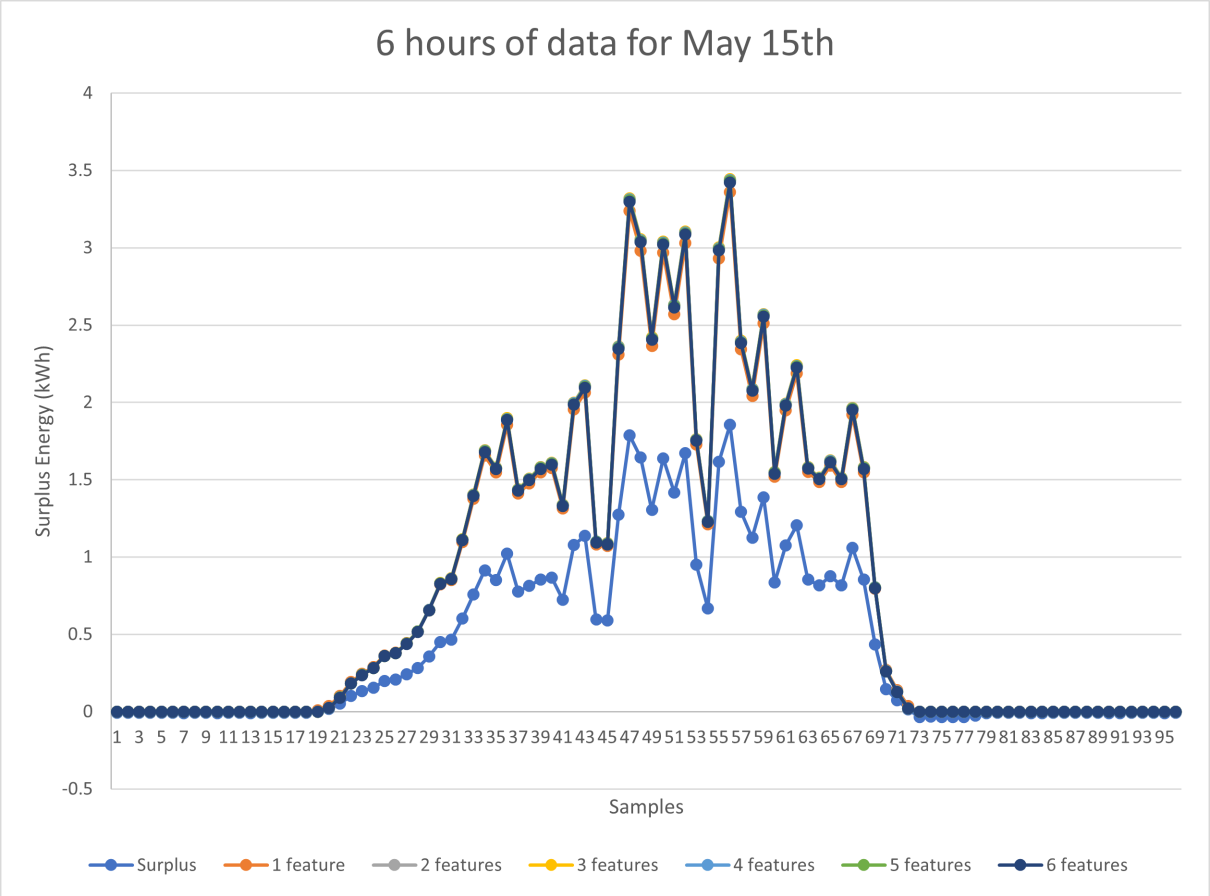


Figure E.2: Graphical representation of forecasts using 6 hours of day 14 to forecast the day 15. Due to the proximity of the values obtained with the models that use 1 to 6 features as X predictor values, the graphical visualization of all the results was impaired.

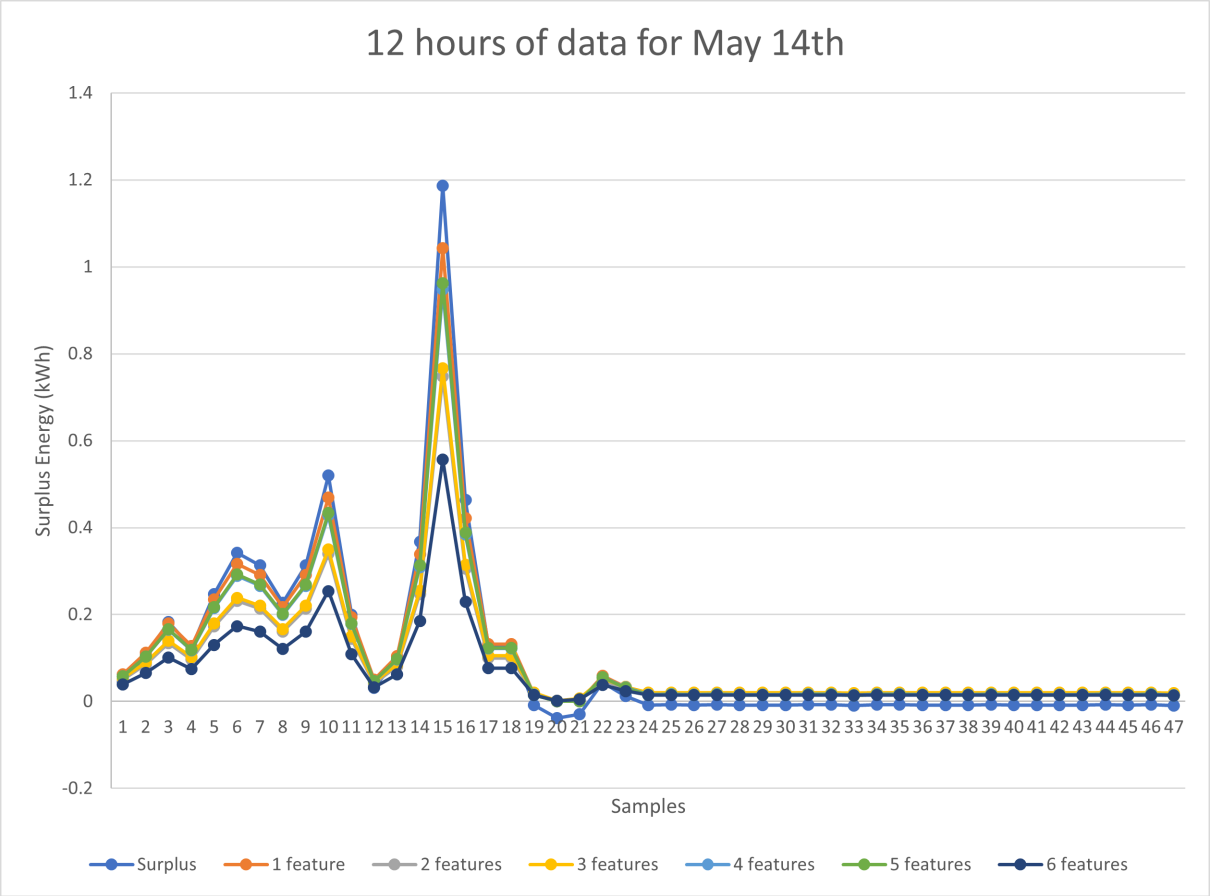


Figure E.3: Graphical representation of forecasts using 12 hours of day 14 to forecast the rest of the day.

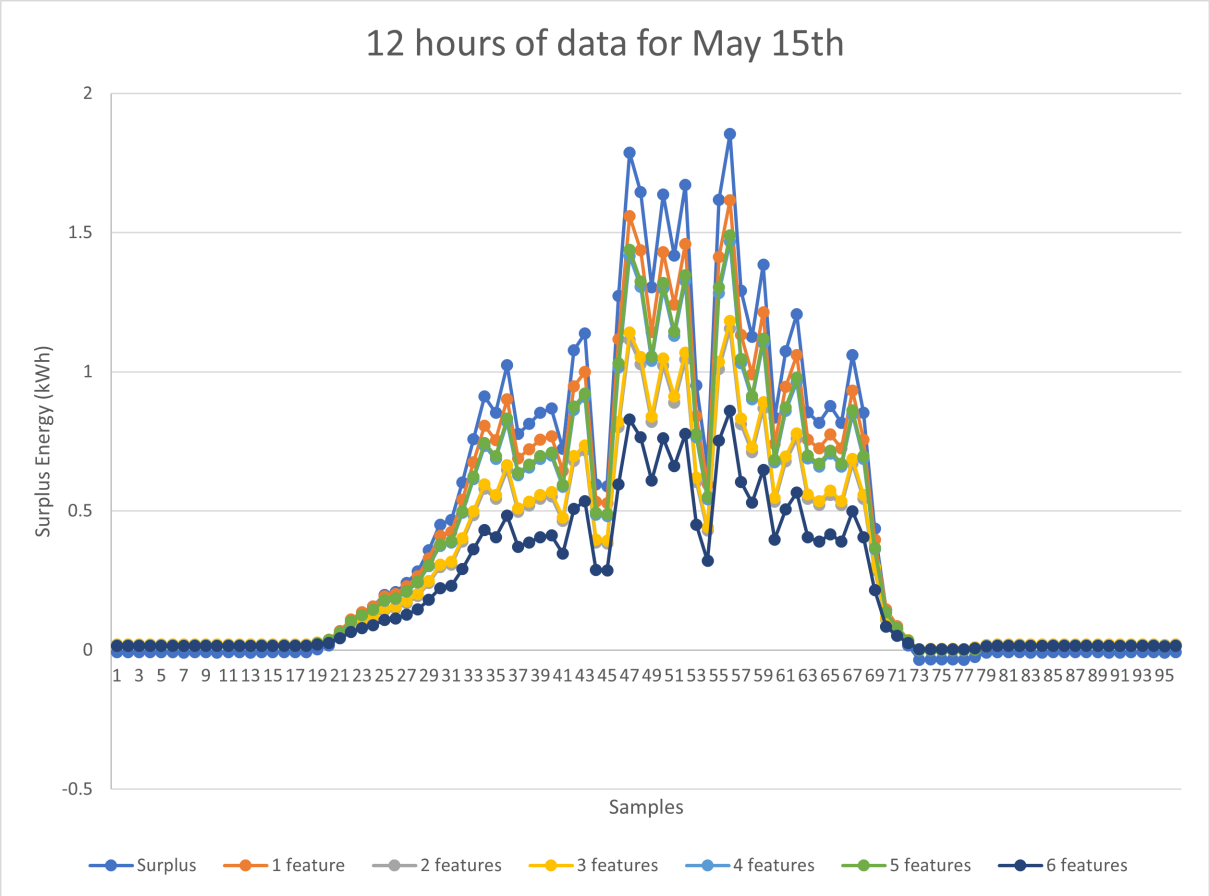


Figure E.4: Graphical representation of forecasts using 12 hours of day 14 to forecast day 15.