iscte
INSTITUTO UNIVERSITÁRIO DE LISBOA

SmartFarm: Improve Sustainability using Wireless Sensor Networks

João Miguel Botas Cardoso

Master in Telecommunications and Computer Engineering

Supervisor
Professor Doctor Pedro Joaquim Amaro Sebastião, Assistant Professor
ISCTE-IUL

Co-Supervisor
Master André Filipe Xavier da Glória, Researcher
ISCTE-IUL

November, 2020

*Aos meus pais Cristina e Mário e ao meu avô João por me terem dado todas as condições para o meu percurso académico, sem eles nada disto seria possivel.*

## Acknowledgment

First of all, I would like to thanks to ISCTE - University Institute of Lisbon for having been my second home in the last 5 years, having been the best years of my life, where, besides all the knowledge that my course has taught me, I have most developed as a person and learned many life lessons.

To PhD Pedro Sebastião for all the opportunities and solutions offered beyond all the orientation and support given throughout the development of this dissertation.

To André Glória for all the support he gave me, taking away all my doubts, resolving all the problems that appeared along the way, never having a "no" as answer and always having the patience to explain things once again. Thanks for being the professional and friend that you are, I'm sure the future will smile on you.

To my colleagues and friends who shared this 5-year adventure with me, especially to João Antão (The Uncle), to Bernardo Brogueira, to the little big twins Inês Gomes and Filipa Gomes, to Gonçalo Simões, to Carolina Dionísio and to Jorge Rafael, for, each in their own way, having marked this academic path by the positive.

To Carolina Fiuza for the love and support at all times, having the patience to perceive me and give me strength whenever I needed it.

Finally, and most important, a big thank you to my family for giving me all the conditions to get here. To my mother Cristina Botas for doing everything so that I don't lack anything, turning the world around if necessary, and for all the constant worries, to my father Mário Cardoso for all the conversations and for having had the courage of, so many years later, also to embark on the adventure that is to make a master's degree. To my grandparents for making sure they are present, even when they are no longer physically present, always having a word of comfort and motivation to give, offering any help that I need, especially to my grandfather João Botas for having always supported my studies since the beginning.

# Resumo

Nos tempos que correm, a poupança de recursos naturais é cada vez mais uma preocupação, sendo a escassez de água um facto que se tem verificado em cada vez mais zonas do globo.

Uma das principais estratégias utilizadas para contrariar esta tendência é o recurso a novas tecnologias. Neste tópico tem se destacado a Internet das Coisas, sendo estas caracterizadas por oferecerem robustez e simplicidade, sendo ao mesmo tempo de baixo custo.

Nesta dissertação foi apresentado o estudo e desenvolvimento de um sistema de controlo automático para rega de campos agrícolas. A solução desenvolvida contou com uma rede de sensores e atuadores wireless, tendo por base estudos anteriores ao nível dos módulos e protocolos de comunicação utilizados, uma aplicação movel para iOS que oferece ao utilizador a possibilidade de consultar os dados coletados em tempo real e o histórico dos mesmos e ainda atuar em conformidade.

De forma a adequar a administração de água, foram estudados algoritmos de Machine Learning que prevejam a melhor hora do dia para a administração de água, dos algoritmos estudados (Decision Trees, Extreme Gradient Boosting (XGBoost), Random Forest, Redes Neuronais e Support Vectors Machines) o que obteve melhores resultados foi o XGBoost, apresentando resultados de precisão de 87.73%. Para alem da solução de ML foi também desenvolvido um script que calcule a quantidade de água necessária a administrar ao terreno em analise.

Através da implementação do sistema foi possível perceber que a solução desenvolvida é eficaz, conseguindo atingir valores de 60% de poupança de água.



**Palavras-Chave:** Internet-das-Coisas, Machine Learning, Agricultura Sustentável, LoRa, Swift, ESP32

## Abstract

Nowadays, the saving of natural resources is increasingly a concern, and water scarcity is a fact that has been occurring in more areas of the globe.

One of the main strategies used to counter this trend is the use of new technologies. In this topic the Internet of Things has been highlighted, these solutions are characterized by offering robustness and simplicity, while being low cost.

In this dissertation was presented the study and development of an automatic irrigation control system for agricultural fields. The developed solution had a wireless sensors and actuators network, based on previous studies at the level of modules and communication protocols used, a mobile application for iOS that offers the user the capability of consulting not only the data collected in real time but also their history and also act in accordance with the data it analyses.

In order to adapt the water management, Machine Learning algorithms were studied to predict the best time of day for water administration, of the studied algorithms (Decision Trees, Extreme Gradient Boosting (XGBoost), Random Forest, Neural Networks and Support Vectors Machines) the one that obtained the best results was XGBoost, presenting results of 87.73% of accuracy. Besides the ML solution, a script was also developed to calculate the amount of water needed to manage the fields under analysis.

Through the implementation of the system it was possible to realize that the developed solution is effective and can achieve up to 60% of water savings.



**Keywords:** Internet of Things, Machine Learning, Sustainable Agriculture, LoRa, Swift, ESP32

# Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| ACK | Acknowledgements |
| ADC | Analog to Digital Converter |
| API | Application Programming Interface |
| CSS | Chirp Spread Spectrum |
| DT | Decision Trees |
| FSK | Frequency Shift-Keying |
| GPIO | General Purpose Input/Output |
| GSM | Global System for Mobile Communications |
| HTML | HyperText Markup Language |
| HTTP | Hypertext Transfer Protocol |
| IEEE | Institute of Electrical and Electronics Engineers |
| IDE | Integrated Development Environment |
| IoT | Internet of Things |
| IP | Internet Protocol |
| IPMA | Instituto Português do Mar e da Atmosfera |
| LoRa | Long Range |
| LTE | Long-Term Evolution |
| ML | Machine Learning |
| MLP | Multi-layer Perceptron |
| MQTT | Message Queue Telemetry Transport |
| NB-IoT | Narrow Band Internet of Things |
| NN | Neural Networks |
| REST | Representational StateTransfer |
| RF | Random Forest |
| SQL | Structured Query Language |

| | |
|---|---|
| SSR | Solid State Relays |
| SVM | Support Vector Machine |
| TCP | Transmission Control Protocol |
| URL | Uniform Resource Locator |
| Wi-Fi | Wireless Fidelity |
| WSAN | Wireless Sensor and Actuator Network |
| WSN | Wireless Sensor Network |
| XGBoost | Extreme Gradient Boosting |

CHAPTER 1

# Introduction

## 1.1. Motivation

Agriculture was always the main supplier of food for society, being responsible for more than 74% [**1**] of the population daily consumption. In order to keep the production for the growing number of world population, changes must take place, mainly when the growth in production also increases the consumption of water. In addition, with water scarcity being a worldwide concern, sustainability and the use of technology in agriculture have become big trends.

With the evolution of technology and the way information is used in our daily lives and activities, where is possible to know any kind of information at any time and place, with a simple smartphone or computer, devices and technology are being implemented in activities that until now have never been used. This evolution is possible due to the great number of small and cheap devices that are connected to the Internet and are capable to collect crucial information. These devices can also combine a wide range of technologies, leading to the creation of Internet of Things (IoT) [**2**].

The proliferation and success of IoT solutions was due to the ability of creating networks of sensors and actuators, technology that had a big evolution in the past years, with the creation of Wireless Sensor Networks (WSN) that, when working in parallel with IoT, can provide the user a numerous amount of functionality and solutions, allowing to connect a large number of IoT devices, send data and control information through online platforms or mobile apps [**3**].

In agriculture, the key point to success is the water administrated to the fields. So, it becomes necessary to monitor the fields to calculate the exact amount of water and the best hour of the day to administrate it. Until recent times, water management was done by humans, mostly in trial and error fashion, leading to complex procedures and inaccurate results. With the use of calibrated sensors and real time information provided by them, this process can be easier and more precise.

When administrating the correct amount of water, not only a natural resource, that nowadays is more and more in danger, is being saved, but also a more sustainable way to create food is applied, with better conditions and fewer cost to the farmer.

## 1.2. Objectives

The goal of this thesis is to develop a system to improve sustainability and efficiency of irrigation in agricultural fields in order to work according to the field needs. Through the collection of real time local data about the field conditions, it is possible to improve the irrigation system, as well as give the owner the appropriate information, like the best time of the day for irrigation. This project allows the farmer to have a better understanding of their fields and reduce the costs of maintenance.

In order to achieve the proposed goal, it becomes necessary to create groups of modules that, when placed together, will result in the final project. These modules are as follows:

- **Sensors nodes:** In order to collect the necessary data from the field such as moisture and temperature from the ground and air, wind direction and speed and rainfall. Not only the use of sensors is a crucial part of this project, but it is also important to choose the right sensors and install them properly, as well as the right microcontroller, that better adapts to the specifications that the project requires.
- **Communications:** In order to transmit the collected data in a reliable and efficient way, a way to connect the nodes within the system and the network to the server is necessary, and for that the best solution needs to be studied and developed.
- **Algorithms:** To analyse all the collected data, a process, that will run on the server, will be developed to analyse all the collected data and perform the necessary actions, based on Machine Learning techniques.
- **Batteries:** Given the variety of nodes that will be used and the location of application, it becomes necessary to choose the right batteries to every node in order to achieve the longer operation life without maintenance and with the possibility to self charge.
- **Application:** To give full access to the user an interactive dashboard will be developed, that will be linked to the server and the database to provide the capability of analyse the collected data and to perform tasks, only requiring an internet connection.

The whole system should consider sustainability, in other words, everything should be implemented in order to reduce consumption of natural and material resources, like energy and water, and reduce costs for the final consumer.

### 1.3. Scientific Contribution

This dissertation presented some contributions, such as:

- The presentation of a responsive agriculture monitor system;
- The development of a Machine Learning (ML) solution capable of predict the best hour of the day for irrigation and a script to calculate the duration of it;
- The development of a Mobile Application capable of establish connection with the system in the field and perform task in a remote way;
- Implementation and study of several technologies that integrate the developed solution and demonstrate their capacities.

Besides the resulted dissertation and the mentioned contributions, this study also contributed to the publication of the following scientific papers in international conferences and journals:

- J. Cardoso, A. Glória, P. Sebastião, "A Methodology for Sustainable Farming Irrigation using WSN, NB-IoT and Machine Learning," in 2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM);
- J. Cardoso, A. Glória, P. Sebastião, "Improve Irrigation Timing Decision for Agriculture using Real Time Data and Machine Learning" in 2020 International Conference on Data Analytics for Business and Industry: A Way Toward Sustainability (ICDABI);
- A. Glória, C. Dionísio, G. Simões, J. Cardoso, P. Sebastião and N. Souto, "Water Management for Sustainable Irrigation Systems using Internet of Things," Sensors (Switzerland), vol. 20, no. 5, pp. 1–14, 2020;
- A. Glória, J. Cardoso, P. Sebastião, "Improve Energy Efficiency of Irrigation Systems using Smartgrid and Random Forest" in 2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM).

## 1.4. Structure of the Dissertation

This dissertation was divided by chapters. After this introductorily chapter, where was described the motivation and the objectives for this study, is presented the State of the Art Chapter, with an overview on the current research status, a review of the used technologies in previous projects and the ones chosen for the proposed system, regarding hardware, communications and software. In Chapter 3 is presented the system architecture, with a description of all the system logic and how the pretended tasks and goals will be performed and reached. Chapter 4, presents the study done on Machine Learning, being presented the followed methodology to achieve the best algorithm that can predict the best hour of the day for irrigation. Chapter 5 explains how the system was implemented, starting from the system setup, where the system was divided in some parts and tested individually, to the real test case, in real case scenario, and the obtained results. Finally, in Chapter 6 conclusions are presented, based on the entire study and work, and future work is proposed.

CHAPTER 2

# State of Art

This chapter presents an analysis of existing resources for the development of this project, with a focus on how Internet of Things (IoT) is proliferating and impacting people's daily lives in all areas. So, in this chapter it will be presented the research made in order to develop the pretended system, being presented IoT and Wireless Sensor Network (WSN) capabilities and the solutions that these technologies can offer when implemented together. Regarding to the enabling technologies, will be discussed how data can be gathered, transmitted and analysed, with a comparison the communication protocols that better adapt to the pretended solution. For each of the topics, some the related work is presented and how we will improve upon them. Finally, it is presented how our work follows the research being done in our group, including all the technologies already studied and used in similar solutions and what we can use and improve to develop our solution.

## 2.1. Internet of Things

Over the past years, Internet has had a huge development, being integrated in solutions for almost every situation. The most recent part of this evolution is the IoT, a new part of Internet technology that brought a new range of solutions and conditions for expansion of Internet in our day-by-day activities [**4**].

IoT has the capacity of collecting data from physical objects, process that data and perform actions according to that. This brings new ways for performing tasks that must be done daily or that need to be triggered automatically when some situation requires it, among many other which leads to the user's well-being [**5**].

In this way, IoT has been introduced throughout our lives, bringing numerous benefits to which we are already accustomed and which are often not identified, being applied in health, helping the management of queues or appointment booking, in smart homes where with just one touch of a mobile application is possible to control the lights, monitor energy consumption or even know if someone is at home. Among many other examples that, through automation and self-learning, IoT performs an immense number of tasks in order to offer comfort and productivity [**5**].

The result of the continuous increase of IoT solutions is the growing number of active and in use devices. In 2009 the number of devices in use was 0.9 billion and was expected to reach 26 billion by 2020 [**6**], however the author of [**2**] indicates that in 2012 8.7 billion devices were connected, with growth to 34.8 in 2018 and a prospect of reaching 50 million by 2020, far exceeding the first predicted figures, and it is possible to conclude that the figures will continue to rise to unexpected values.

In order to better understand in what IoT consists, it can be divided in six main components (Identification, Sensing, Communication, Computation, Services and Semantics) [**2**], for its implementation, IoT follows the system presented in Figure 2.1 as a model.

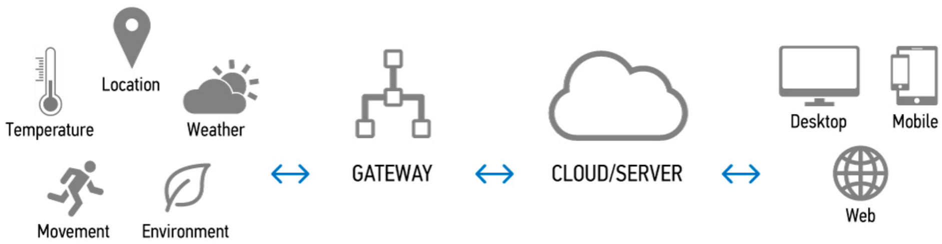


FIGURE 2.1. IoT Implementation Model

The way that components are integrated, as it will be described ahead, can vary from solution to solution, based on the needs and specifications of each project, but the main core of an IoT solution will be always gathering information, transmit them to servers and process them using smart computation, in order to create outputs that benefit the final user or improve the task they are handling.

## 2.2. Green Tech and Sustainability

These days, concerns about climate change, ecological footprint, water shortage, among many others, are increasing and technology has moved in the direction of creating solution that help reduce these impacts on the planet, with this so called "Green Tech".

"Green Tech" works to explore new solutions that do not contribute to the situations presented above and seeks to replace the most polluting and consuming technologies with those that are not. This demand, as far as hardware is concerned, is based on the creation of modules with lower consumption and which do not compromise their performance, or the use of building materials based on recycled material, being one example the creation of modules with deep sleep capabilities, a feature to be implemented in microcontrollers that make it possible to put the whole system it controls, like sensors, into an energy saving mode, consuming as minimal energy as possible. As far as software is concerned, the demand for logic that leads to the same results using less resources, for example energy, the demand for solutions that can reduce energy consumption at standby times or even algorithms that prevent excessive consumption of natural resources [**7**].

Green Tech are, in these days, improved and been implemented in many areas, being agriculture one of the areas where the awareness for climate changes is a big concern. This way, solutions have been searched and developed to go against this trend.

### 2.2.1. Agriculture Application

Regarding to agriculture applications, IoT has proved over the years to be one of the main ways of achieving sustainability, given the number of solutions it offers, which can be applied in various areas of agriculture, such as controlled planting, open field planting or livestock raising. With the application of these or other areas, is possible to reduce energy consumption or improve the quality of agricultural holdings, having all these factors an economic impact [**2, 8**].

Besides that, one of the highest concerns on agriculture application is the waste of natural resources, such as water, since typical irrigation systems do not concern with the control and monitoring of irrigation devices such as water pumps.

Situations such as these can lead to errors or breakdowns which, when not prevented, can lead to water leaks that, not only represent an elevated waste of water, but can also lead to flooding and crop compromising. In order to prevent situations like these, software that is constantly looking for failures and that has the mechanisms to solve them immediately must be developed, being these also another form of "Green Tech".

### 2.2.2. IoT and Green Tech for Sustainable Agriculture

With the advances of technologies over the years, mainly in IoT, a huge range of solutions that create options to improve field monitorization and agricultural results has appeared. The creation of low cost, low power consumption and ambiguous solutions are one of the most researched and developed projects.

In [**9**], the author intended to develop an IoT solution with the aim of achieve sustainability and offer the user a cheap solution. The developed system was designed to be installed in agricultural fields and proceed to his monitorization in order to understand its conditions and the water administration needed. This system consisted in a WSN with a star typology, having one center node where all the collected data were sent and many sensor nodes, having each one a single sensor attached to him. Since for proper monitorization several types of sensors are needed, using one sensor node for each sensor leads to a huge number of nodes on the network, which can lead to a network overload when the system needs to be implemented on larger grounds. Another point that was not considered was the need to introduce more sensors into the network, since the developed system to control the network was implemented in a static manner.

The author of [**10**] developed a system for precision irrigation for large fields, so it was necessary to use a high number of sensors, 15 000 in some cases. The developed system was created thought a WSN and a central server where all the collected data was stored and the necessary calculations were performed. The presented solutions where quite complete since, in addition to the fields monitorization, it also focused on water reserves, water distribution and water consumption. However, the huge amount of data exchanged and processes running lead to a system overload which compromised part of the system. Besides that, the proposed system needs a special configuration to adapt to each proposed scenario, making the system less adaptive and more complex.

In [**11**] is presented a smart agricultural solution through IoT technologies. The presented study had as focus the use of IoT solutions to offer the user a good management experience through the implementation of a set of sensors in the field and communication modules creating a WSN. The collected data by the sensors was sent directly to the data centers where all the data was stored and then presented to the user through a mobile application. This system presented a monitorization solution, allowing the owner to monitor his crop without the need to go to the site, however there was no solution to place any action automatically or manually to counter any extreme field situation like lack of water when it was necessary to turn on the water pumps.

Regarding to Machine Learning (ML) solutions, there were found some studies about this technology applied to sustainable irrigation. The author of [**12**] presented a high precision agriculture study through the study and application of Decision Trees algorithms to monitor agriculture fields, in this case vineyard. The presented study proposed a data collection only through images captured by cameras installed in drones. The author obtained accuracy results of 90% for most of the performed tests, however there were only studied and tested one ML algorithm and there was not presented any action to be performed according the obtained results from the developed algorithms.

Although several solutions already exist in both academic and industry level, no solution was found that provides all the features intended for this project, allowing our proposal to improve and innovate based on a field and market need.

### 2.2.3. Green Tech and IoT Research at ISCTE

A study group lead by Professor Pedro Sebastião and Researcher André Glória already developed some methodologies for using IoT and WSN to improve the sustainability of day-by-day activities, contributing to several thesis and research papers published in International Conferences and Journals.

The author of [**13**] conducted a study for the creation of a smart house with the main objective of monitoring and identifying energy waste. For his study it was presented a solution based on a WSN, divided in several nodes having each one specific characteristics for the tasks assigned to them and a central server where a database and a control platform were hosted, all of this being controlled by a mobile application for Android. For the implementation of the proposed system, the author made a research at a hardware and software level in order to understand the solutions that better adapted to his model, having concluded that for communication between the network, the LoRa protocol had the best features, taking into account its relation between range and power consumption, in addition to its ability to confirm messages reception through Acknowledgements (ACKs). Regarding the design and control of each node, the author identified that the ESP32 microcontroller had the best specifications when taking into account its processing capabilities, having a dual-core processor and regarding power consumption, with a deep sleep mode, which makes it possible to reduce consumption on a large scale during the time when it is not performing any function. With regards to the exchange of messages with the server is concern, the author also made a study where concluded that the protocol that best suited to this purpose would be the Message Queue Telemetry Transport (MQTT).

The results obtained show that the chosen technologies, both in terms of hardware and software, corresponded to what was intended, and allow for the detection of energy waste situation in households, such as lights or appliances left working while no one was in that room, alerting the user for those situations and thus reducing the waste of energy.

Another project [**14**] used the same implementation and design described above, applying the solution for water management in swimming pools, monitoring the entire environment and checking for situation such as drops in temperature, pH or chlorine levels, among many others, that in a remote way helped the maintenance teams to have a better understanding of what is happening in the pool in real time. Once again, the developed and implemented methodology proved to be reliable and efficient.

As such, our work will take into consideration the previously done research, on both hardware and software and also the complete methodology on how to create a remote monitoring system using IoT. The goal is not only to guarantee that the design and methodology can be applied to other activities, but also to improve the solution with the addition of more advanced technologies, learning systems and dashboard functionalities, such as a NB-IoT communication solution, to improve the implementation of the system in remote areas, data analysis through ML algorithms, and the implementation of a self-generating power supply, through solar panels, to help the power consumption of the system and ease the implementation and scalability, creating a solution that can really be adapted to every scenario.

## 2.3. Enabling Technologies

As previously said, an IoT system is composed by a set of components, that together create a system capable of gathering, transmitting, analysing and act upon an environment. For each of those components, several technologies, on both hardware and software, have been developed in the last years, with new solutions appearing every day, as IoT gains more and more space as a research for both academic and industry.

The following sections explain the importance of each of these components and the available technologies that make them possible and that will allow us to create the proposed system. As explained in Section 2.2.3, the proposed system will be based on solutions already implemented in this area and from which some technologies have already been studied and tested. Thus, those same strategies were followed, with the technologies used being presented, as well as new solutions that cover some flaws discovered in them.

### 2.3.1. Sensing

WSN offer to IoT the capacity to implement a huge number of sensors which can be connected among each other. In the past, this connection was made by wires which confers a limitation to the entire system, not enabling to locate the nodes in the pretended location. Nowadays WSN offers the possibility, through wireless connections, to establish the necessary connection between the nodes, leading to the system robustness, knowing that WSN is designed to be the most efficient possible.

In order to create a structure within the network, nodes are created to each specific task. Theses nodes can be divided in three categories [**2**] , as shown in Figure 2.2.

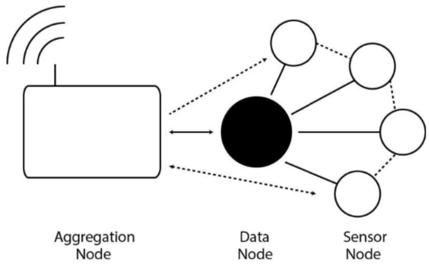


FIGURE 2.2. Sensor Network Nodes

In a first layer are the Sensor Nodes, the simplest node in the network, and as such the ones at the lowest level of a WSN. Normally composed only by microcontroller, a set of sensors and a communication module, their only task is to collect data from the sensor installed and send it to another higher node.

In a second layer are the Data Nodes, these nodes are responsible to receive the collected data from the Sensor Nodes and send it to the Aggregation Node, working as a middleman. For simple and small solutions this node does not necessarily have to be implemented, however for more complex solutions his implementation is crucial to ensure the correct management of the network.

In a final layer is the Aggregation Node, a node without sensors connected, being responsible for receiving data from the other nodes and sent those data to the main server, since this is usually the only node with Internet connection.

As mentioned before, Internet has evolved over the years and IoT has undergone the same trend and to these typical WSN constitution was possible to add a fourth node, the Actuator Nodes [**15**] leading to Wireless Sensor and Actuator Network (WSAN), Figure 2.3. These ones are responsible to perform tasks to stabilize specific values in order to reach the required ones. These tasks can be sent from the server via the aggregation node, since the Actuator Node is always listening for new messages with tasks created after the main server's data analysis. In order to achieve the processed goal, these nodes are equipped with actuators that are responsible to perform the required tasks, such as open or close valves, control lights or motors.

All of the capabilities described above are possible due to the already mentioned evolution of IoT solutions, being one of the main evolutions, the microcontrollers. These modules constitute the center of the processing of all the collected information, sent or

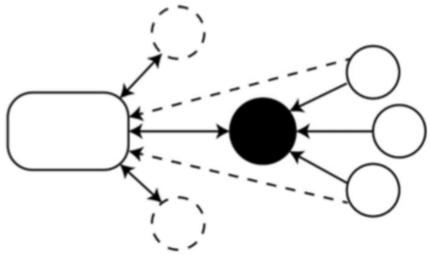


FIGURE 2.3. Sensor and Actuator Network Nodes

received, making possible the creation of the network. These microcontrollers installed in each node, due to their capacities, are responsible for the control of all the modules connected to them. Furthermore, these microcontrollers are already developed in order to offer simplicity and high performance with a lower cost. The available solutions in the market of microcontroller are wide, with specific offers for specific needs, from high computational projects to low cost ones. Typical solutions found in the research world, for the low cost projects, include the Arduino Uno, ESP32 and Raspberry Pi, all design for the common user, with simple interfaces and ways to program them, working with almost any sensor or actuator.

Although Raspberry Pi offers higher processing capacities, its power consumption and size are disadvantages for the intended functions and installation locations. As far as Arduino Uno is concerned, it has a single core processor, although it is smaller in size and consumption than the Raspberry Pi, it is still larger compared to ESP32 and none of it has deep sleep mode [**16, 17, 18**].

TABLE 2.1. Microcontrollers Characteristics

| **Features** | **Raspberry Pi 3** | **Arduino Uno** | **ESP32** |
|---|---|---|---|
| **Dimentions(mm)** | 85 x 57 x 17 | 68 x 63 x 17 | 54 x 28 x 10 |
| **Integrated Communication Protocols** | UART, SPI, I2C WiFi, BLE | UART, SPI I2C | UART, SPI I2C, WiFi |
| **Programming language** | Any language | C# | C# |
| **Deep-Sleep Mode** | No | No | Yes |
| **Power Consumption [mA]** | 300 | 50 | 80-260 |
| **Complexity** | High | Medium | Medium |

So, after analysing the characteristics of each of the microcontrollers, presented in Table 2.1, considering the study done and the methodology presented by the previous works, as presented in Section 2.2.3, from which we are taking some guidelines to create our project and taking into account the specifications necessary for the application of the proposed system, it was possible to realize that the ESP32 is the microcontroller that better adapts to it, due to its dual-core processor, its performance and energy consumption ratio and its deep sleep feature.

**2.3.1.1. *ESP32.*** The ESP32 microcontroller, Figure 2.4, developed by Espressif System, is a board with high processing capacities, having a dual-core processor, low power consumption, GPIO interface that allows the control of the boards that it needs to connect. In addition to the low power consumption already mentioned, this board has also a deep-sleep capability, which allows to lower the power consumption even more when it is not in use.

In order to connect and control peripherals, this model has a GPIO of 32 pins, being 12 of them analog pins. Other characteristics, that makes it very attractive for IoT solutions, are the dual-core processor and small dimensions, that put him easily in a node case with the controlled peripherals [**16**]. It has a 2.4GHz Wi-Fi and a Bluetooth connection capability with a flash memory of 2MB and operates from 2.2 to 3.6V [**16**].

In terms of consumption, this model has a low power consumption and deep sleep mode [**19**]. This feature makes it possible to largely reduce the power consumption of the circuit, in which the microcontroller is inserted, during the time intervals in which it is not being used, for example between data collections. Thus, this characteristic becomes vital for circuits that are powered by batteries or solar panels, leading to an increase of the duration of these and reducing the need for system maintenance.

Figure 2.4. ESP32 module

### 2.3.2. Communication Protocols

As mentioned in Section 2.1, IoT systems requires communication in order to achieve the proposed goals. To do that, it becomes necessary to study several communication protocols capabilities in order to choose which one better adapt to the proposed system.

Wireless Network Technologies are one of the most used communication protocols in IoT and are the ones that better fit to the proposed solution. There are many wireless communication protocols that can be used in IoT, such as Wi-Fi, Bluetooth, Zigbee, LoRa or NB-IoT which are the ones that will be analysed in detail in the next sections. In Table 2.2 is possible to analyze the key characteristics [**20, 21, 22, 23**] of each one:

TABLE 2.2. Wireless Protocols Characteristics

| **Features** | **Wi-Fi** | **Bluetooth** | **Zigbee** | **LoRa** | **NB-IoT** |
|---|---|---|---|---|---|
| **Frequency(GHz)** | 2.4 | 2.4 | 2.4 | 0.433 | 1.8 |
| **Range[m]** | 1-100 | 10-100 | 10-100 | 2000 | - |
| **Nodes** | 32 | 7 | 65540 | - | - |
| **Topology** | Star | Piconet | Star/Mesh | Star | Star |
| **Power Consumption [mA]** | 100-350 | 1-35 | 1-10 | 1-10 | 10-100 |
| **Complexity** | High | Medium | Medium | Low | Medium |

A study already done by [**13**] and [**14**], showed that the communication protocol that best suits intra-network communications, for the exchange of messages between each node installed in the field, is LoRa. This choice, and since it is not possible to perform tests in the final system, was made according to the criteria described by the author of [**24**] where was explained that the chosen protocol should guarantees the following criteria: Inter-operability, Self-Management, Maintenance, Bandwidth and Power Consumption. So, given its long range, low energy consumption and low complexity, this protocol stands out from the other ones.

Regarding the communication between the aggregation node and the server and mobile application, according to the study made by [**13**] and [**14**], it is possible to understand that the protocol that best suits the proposed system is the MQTT. Since this protocol was developed to be applied in IoT solution due to its low complexity.

The MQTT protocol requires an internet connection, which can be established thought the Wi-Fi capabilities of ESP32, however agricultural fields are not characterized by Wi-Fi coverages which leads to other solutions that enable an internet connection like NB-IoT.

**2.3.2.1. *LoRa.*** LoRa, meaning Long Range, is a bidirectional communication protocol which provides low power long range communications. In order to be capable of that, LoRa is based on Chirp Spread Spectrum (CSS) modulation which have the same power save that Frequency-shift Keying (FSK) modulation (FSK modulation is one of the most common on wireless communication protocols) but with a significant increase in terms of communication range [**20**]. These characteristics in combination with the star topology used, which gives a lower complexity network, stretch the capability of support 2000 meters connections with a low power consumption [**25**] which makes it very attractive for IoT systems since most of them work from batterie supplies.

**2.3.2.2. *MQTT.*** Message Queue Telemetry Transport (MQTT) is an open protocol developed by IBM [**26**] based on TCP/IP protocol. This technology was developed in order to match the IoT requirements, due to his low complexity and need of internet bandwidth. MQTT works in a publish/subscribe paradigm [**27**], being based on two actors, publisher and subscriber. Publisher create messages and attach it to a specific topic. Any interested subscriber on this topic will subscribe it and receive all the messages published in that topic. The topic is identified by a number called topicID. All the players can be subscribers or publishers [**27**].

**2.3.2.3. *NB-IoT.*** Narrow Band IoT (NB-IoT) is a technology released by 3rd Generation Partnership Project (3GPP) in June 2016 [**28**]. This technology can coexist with Global System for Mobile Communications (GSM) and Long-Term Evolution (LTE) under licensed frequency bands. NB-IoT has enhanced IoT in terms of coverage, power saving capabilities and complexity reducement, giving to the system the capability to establish internet connections through mobile network [**23**].

### 2.3.3. Data Analysis

As described before, IoT has had a great evolution in the last few years, which leads to the increase of data created, transported and stored. This new reality created a new necessity, in order for projects to act according to the data they collected. It was necessary to use technologies with the capacity to process it in real time.

This high number of information collected established new challenges, considering the volume, the wrong or useless data or the need to consider this data in future actions. With the impossibility to analyse each value manually, due to the volume, the need to act in a rapid way and the need to create knowledge that can be used in future actions instantaneously, technology needs to be applied. Also, some data is characterized by containing some noise that must be identified and eliminated before the analysis [**29**]. Given this challenges, ML arises as the needed technology, as it has the ability to analyse big volumes of data in little time and create knowledge from them.

ML offers solutions for the previously exposed problems, creating learning process that automatically improve over the time, since this technology improves as more data is added to the algorithms. In addition, the trained ML algorithms provide a constant processing of new data, giving results in real time, which is a fundamental feature for any project.

Since this technology include the ability to learn and respond in real time, the accumulation of data is certainly an aspect to take into account as this increase processing capacities on a large scale. For this, ML contains a vast number of algorithms based on mathematical models that are currently under high development and that, through their study and adaptation to the model to which it will be implemented, also constitute a key feature for the solution.

In order to achieve the pretended ML algorithm, there are some learning techniques that can be used, among which are the most used which are supervised learning and un-supervised learning. Supervised learning is one of the most used techniques to develop ML algorithms, this technique is based on providing samples (entries) in a dataset where, during the training process, is provided to the developing algorithm both the input as the output(the one that the final algorithm as to predict after the training phase). Each entry of the dataset is characterized by an established number of features, being the same for all entries. Regarding to un-supervised learning, this technique is known as the process of the training without the knowledge of the complete dataset, being the training process focused in data discovers and the find of hidden patterns [**30**].

The supervised learning is divided in classification and regression. Regression achieve the pretended values $Y$, through the given features $X$. The followed mathematical notation is shown in Equation 2.1 where is presented the notation for a linear regression.

$$\mathrm{Y} = \mathrm{f(X)} + \varepsilon \tag{2.1}$$

In Equation 2.1, $Y$ represent the dependent value (output), $X$ the independent value (input), $f$ the function that establish a relation between both and $\varepsilon$ represents the error [**31**].

In regards to classification methods, this one is known for its goal of approximate a mapping function from the inputs given by the dataset, in order to identify the output values. In Equation 2.2 is shown the followed function by this method, where f is the mapping function, $X$ the input value and $Y$ the output value [**32, 33**].

$$y(f : x \rightarrow y) \tag{2.2}$$

Taking into account the equations presented before, it was possible to understand that there are same similar points in the presented methods, however there are some significant differences since classification is developed to predict a range of classes such as true or false, or 1 or 0, respectively, or even a hour of the day while regression methods were developed to try to predict a range of numbers that are not predefined, for example a price of a car. Thus, and considering the characteristics of the two presented methods, classification methods were used.

**2.3.3.1. *Random Forest.*** Random Forest (RF) is a tree-based method that conglomerates several self-determining decision trees developed for classification and regression. To each tree is given a portion of entries of the dataset given to the algorithm during the training process. Through the combination of the various trees this algorithm is able to understand which is the best option through the result of each one [**34**]. Figure 2.5 shows an example of Random Forest operating logic.

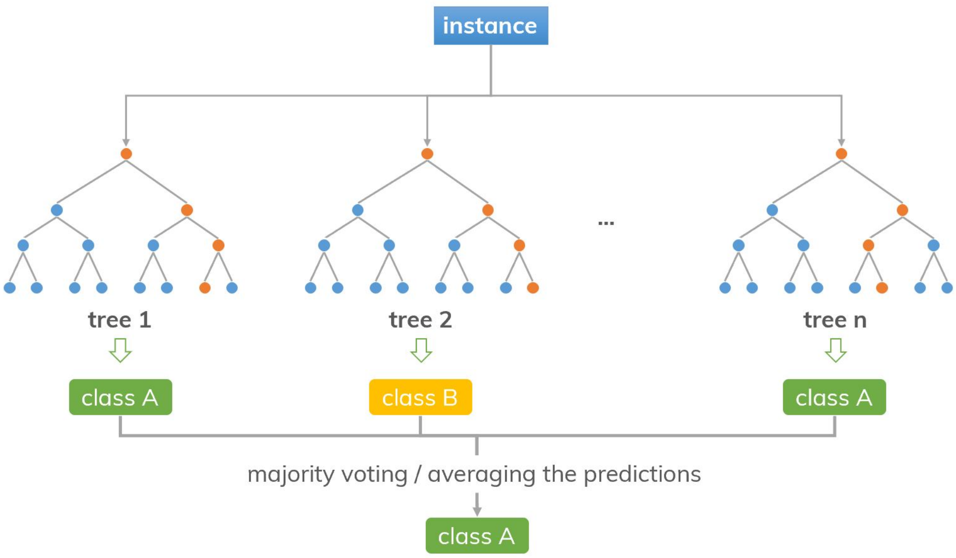


FIGURE 2.5. Random Forest Working Logic

**2.3.3.2. *Decision Trees.*** Decision Trees (DT) are tree-based methods in which each path begins in a root node and multiple divisions are made, taking into count the dataset, creating sub-trees, representing a sequence of data divisions until it reach a leaf node with an outcome. These methods can be applied for classification and regression. The final goal of this method is to reach a model that can predict the search value for that specific scenario by learning simple decision rules [**35, 36**].

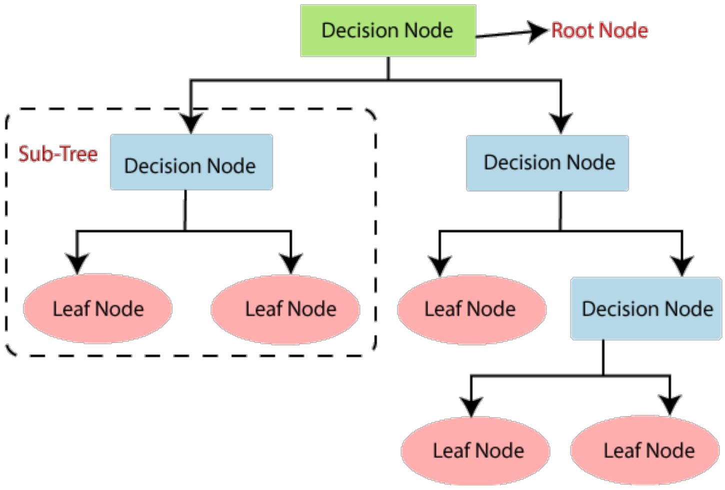


FIGURE 2.6. Decision Tree Working Logic

**2.3.3.3.** ***Support Vector Machines.*** Support Vector Machines (SVMs) are a set of supervised learning methods developed for classification and regression which are known by his high effective in high dimension spaces and for its use for training points in the decision function, being also memory efficient. The working strategy of this algorithm is to create a $n$ dimension point between the classes under study in order to establish a division mark. The operation logic can be seen in Figure 2.7 [**37**].

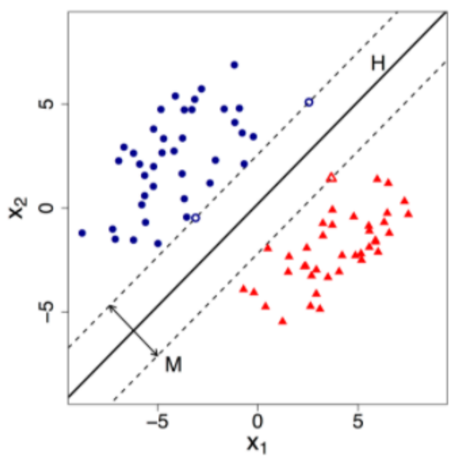


FIGURE 2.7. SVM Working Logic

**2.3.3.4.** ***Neural Networks.*** Neural Networks (NN) algorithms are defined as computational models of neural system composed by several neurons connected one to the other by synapses, these ones are used to transmit information to the neurons nearby, being the receiver neuron responsible to process the received information and send it to the next one. This process continues until a final output is found. To implement this technology it was used Multi-layer Perceptron (MLP), which is a supervised learning method that learns a function $f(.) : R^m \rightarrow R^o$ by training on a data set, where $m$ is the number of dimensions to input and $o$ the number of dimensions for output. These MLP networks are characterized by being general-purpose, flexible and non-linear. Their complexity can be changed according to their application by varying the number of layers and units of each layer. In Figure 2.8 can be seen the implementation of this algorithm [**38**].

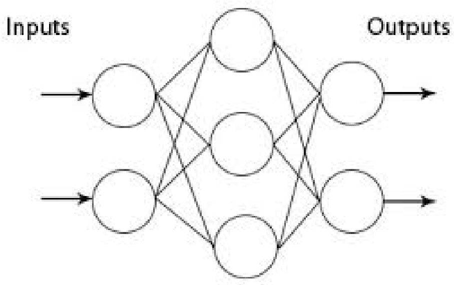


FIGURE 2.8. MLP Working Logic

**2.3.3.5. *Extreme Gradient Boosting.*** Regarding to Extreme Gradient Boosting (XG-Boost), this is a boosting tree-based the implementation of high-performance gradient boosted decision trees. This method takes the decision trees logic to another level, taking into count that gradient boosting methods takes an interactive approach during the training processes, allowing to implement multiple ensemble techniques and, in the end, compare them among each other in order to obtain the best result possible. Considering that the linear combination for multiple trees capabilities that can well fit the training data and describe the complex non-linear relationship between input and output data, makes this method considered one of the best methods in statistical learning [**39, 40**].

CHAPTER 3

# System Architecture

The main goal of this thesis was to develop a system capable of manage agricultural fields through a sustainable and low maintenance solutions. In order to achieve the proposed goal, the system must have a WSN capable of collecting the necessary data and sent it through LoRa and MQTT to the server. This same WSN had also to have the capability of receiving messages from the server to perform the necessary tasks, such as turn *ON* the water pumps during the required time. Besides the messages exchange capacity, the system had to include a mobile application that allows to consult the collected data and manage the field remotely.

As mentioned in the previous chapter, there has been an analysis about the already research and development made on this area and it was concluded that although some similar systems have already been developed, there are some technologies that can be used to make the system even more complete and versatile and with a reduced need for maintenance. So, the mentioned studies were considered for the creation of a new IoT solution, following the same module that can be seen in Figure 3.1.

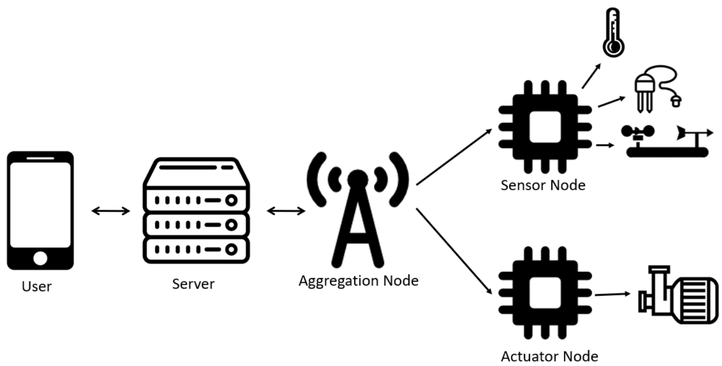


Figure 3.1. System Architecture

## 3.1. Wireless Sensor Network

To achieve the main purpose of this thesis, there was a need to collect data locally, in real time, send that information to the server, without losing any package, analyse all the collected data and create actions that will be held directly on the field.

With the rapid proliferation of IoT, as explained in Chapter 2, solutions like these are more common nowadays, and as explained previously, the proposed work was a continuation of a set of thesis in the IoT and Sustainability areas, which already presented a solution for gathering, transmitting and analysing data, in order to improve efficiency of tasks. Although their work fits for a specific purpose, like households or swimming pools, when porting their research and development to the agricultural fields, some limitations and improvements were found. That allows us to have a proved and functional hardware basis to start, while allocating all our efforts on improving and adapting it to our needs.

So, and keeping in mind the work done previously, it was necessary to create several nodes, each one for a specific task, that together create a WSN, with a star topology, allowing them to share the necessary information among each other.

To achieve the proposed goal, the system needs to include several Sensor Nodes, for data collection, an Actuator Node, to turn *ON/OFF* the irrigation system, and an Aggregation Node, to manage the network and send/receive messages from the server.

### 3.1.1. Methodology

As presented by [**13**] and [**14**], that follow a similar path to construct WSN nodes for their work, although each node has distinct characteristics, there are certain specifications that are common to all, such as the microcontroller and the communication module that provides the communication within the field network. Therefore, all the nodes share a common base, composed by an ESP32 microcontroller and a RFM95W LoRa transceiver.

Regarding to the communication module, which ensures the interaction between all the nodes of the network, it was used the LoRa protocol, given its low energy consumption, long range, low cost of the devices and being a protocol that supports bi-directional communications. The RFM95W module allows long-distance communication with a low consumption relation, being able to transmit up to 2km away with just 70mA.

Not only the hardware already used was taken into consideration, being all the software previously used also studied and adapted, with new scripts being developed, using the C# programming language, which were loaded into the ESP32 installed on each node. For the same nodes was used the RadioHead library to control the RFM95W module. In order to allow the assignment of addresses to each node and allow the message reception confirmation, the RHReliableDatagram.h library was used. With the use of this library, the code was developed to ensure the reception of all messages through confirmation by acknowledgments (ACKs), if the sender node does not receive the ACK and occurs a timeout, time that the sender waits to receive the confirmation, the message was sent again until receiving the respective ACK, with a maximum number of attempts being set and when reached the message is discarded.

**3.1.1.1.** ***Limitations.*** Although the high quality and reliability presented by the system developed by [**13**] and [**14**], some limitations were found, mainly due to the indoor application of the systems:

- The sensor node was powered by alkaline batteries, as they can be easily changed by maintenance in an indoor facility;
- Wi-Fi was used in the Aggregation Node to communicate with the server, as indoor facilities provide connection.

But also, some limitations in terms of hardware and features were found:

- The relay system used in the Actuator Node is not suitable for the irrigation pumps, as they are AC powered, not DC;
- There is no alternative solution for server communications, besides the integrated Wi-Fi on the ESP32.

The core system from these works will be used but considering the goal and implementation of our system, to improve these limitations some improvements and innovations will be developed and implemented in the system, mainly:

- Solar power batteries for the Sensor Node, since they will spread outdoors through large agricultural fields, they can be auto sustainable;
- Alternative communication protocols, mainly cellular, since agricultural fields can be in remote areas and have low connectivity;
- Use of Solid-State Relays that can work alongside the irrigation system.

Alongside the hardware changes, also the software scripts will be adapted to work with the new improvements.

Bearing in mind the core system developed by [**13**] and [**14**] and the limitations found in their work, a new system was developed combining the best features, already described, and some new functionalities. A set of nodes was developed, in order to fulfil the goal of gathering, transmitting, analysing and act upon the environment, in a WSN fashion. Each node, their constitution, responsibilities and features were described in the following sub-sections.

### 3.1.2. Aggregation Node

The Aggregation Node is the one responsible for exchanging data from the other two types of nodes of the network and subsequently with the server.

As explained above, the core of the node is composed by an ESP32 and a RFM95W LoRa module, to enable the exchange of messages among the nodes in the network, and to communicate with the server MQTT was used.

In order to use MQTT, an internet connection was required and, for this purpose, a cellular solution was the best option. In this case, NB-IoT technology was used, since coverage in Portugal is increasing and the technology was developed especially for situations like this, to send small packets of data via a mobile network, allowing small devices to establish internet connection anywhere in the world. To use NB-IoT, the SIM7000E module was attached to the aggregation node, allowing the use of mobile network to establish the internet connection, having the capacity to work through NB-IoT or 2G technologies. The fact that this module can work using 2G brings advantages as agricultural fields are characterized by low network coverage. This technology was one of the improvements made to the system by which the developed system was based, allowing a better adaptation to different environments and making the system more robust.

The constitution of the Aggregation Node can be seen in Figure 3.2.

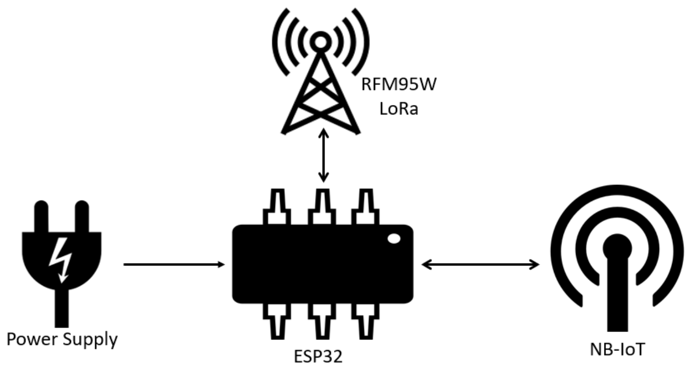


FIGURE 3.2. Aggregation Node

For this node it was necessary to develop a script to control the modules connected to him. To control the NB-IoT module, SIM7000E, was used the Adafruit_FONA_MQTT library, which allows to establish connection with the mobile network and later subscribing and publishing MQTT messages.

**3.1.2.1. *Communication.*** Taking into account that the Aggregation Node was the node responsible for establishing connection between all the nodes in the network, and later exchanging the necessary messages with the server, it was necessary to subscribe one topic and publish it in two MQTT topics.

The subscribed topic, *"thesis/AC"*, was the channel used to send messages from the server or the mobile application to the actuator node whenever it was necessary to change the status of an actuator. This message contained the type of node for which the message was destined, Actuator ("*A*"), the id of the destination node, the nickname of the actuator in the destination node, the state for which it would change (*ON* - 1 or *OFF* - 0) and, if the message indicated that the actuator would be *ON*, the time it would be in that state. The message format was the following: *"Type:NodeId:nickActuator_state_time"*. The Aggregation Node processes the received message and send it to the Actuator Node with the indicated id through LoRa.

Regarding the topics used to publish messages, these were used to exchange information about actuators and sensors. To exchange messages about the actuators the topic *"thesis/BD"* was used. This topic was used to communicate with the server in order to update the status of each actuator whenever it was changed. As explained before, the server sends to the topic *"thesis/AC"* a message whenever it was necessary to change the state of an actuator however, and in order to avoid errors and waste of water, after the change of state it was necessary to send a confirmation message in order to confirm the change of state and to update the database, as well as when the irrigation time as ended.

The second topic, *"thesis/data"*, was used to send the data from the sensor node, with the following structure: *"Type:NodeId:nickSensor_data;nickSensor_data;..."*.

The *"type"* indicates the type of node from which the message came, Sensor (S) or Actuator (A), the *"NodeId"* indicates, as the name indicates, the id of the node from which it came, the *"nickSensor"* indicates which nickname was assigned to the sensor inside each node and finally the value collected by that sensor.

### 3.1.3. Sensor Node

The Sensor Nodes, as the name indicates, are the nodes where the sensors are connected and the responsible for collecting field data. These nodes need to have the necessary sensors to collect the data, a LoRa radio module in order to send the collected data to the Aggregation Node, a control board and a power supply.

The ESP32 microcontroller is used to control all the modules, just like the Aggregation Node. In this node, unlike the Aggregation Node who needs to be always listening, it is possible to use the ESP32 deep sleep functions in order to save energy since the collection of values is timed and the node can be in deep sleep mode during the time of waiting between collections. To send the collected data to the Aggregation Node via LoRa a RFM95W radio is used as explained before.

To power the nodes, due to its low power consumption, it becomes possible to use batteries. In order to make the system more complete and reduce the need for maintenance, a solar panel was also used to charge the battery, ensuring that they will always have enough energy to power the node even during long periods of absence of sun exposure.

In order to ensure the batteries charge it was used the charger module TP4056. The used batteries gives to the system 3.7V, however, as the node is powered by 3.3V a voltage regulator, the MCP1700-3302E, was used to ensure the drop from 3.7V to 3.3V. Also, in order to prevent and smooth the voltage peaks, it was also used a 100 $\mu$F electrolytic capacitor and a 100nF ceramic capacitor. The solar panels circuit and the Sensor Node constitution is shown in Figures 3.3 and 3.4.

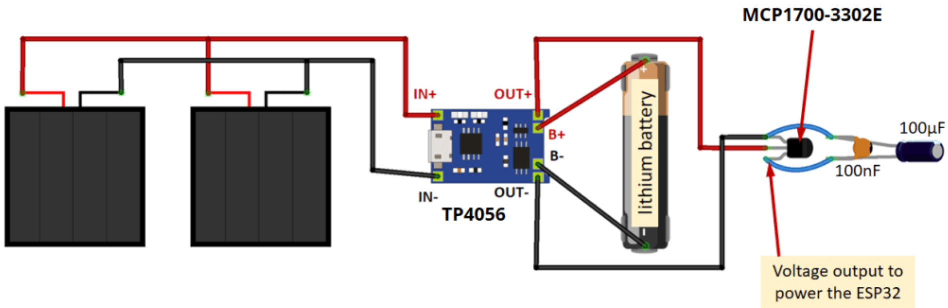


FIGURE 3.3. Solar Panels Circuit

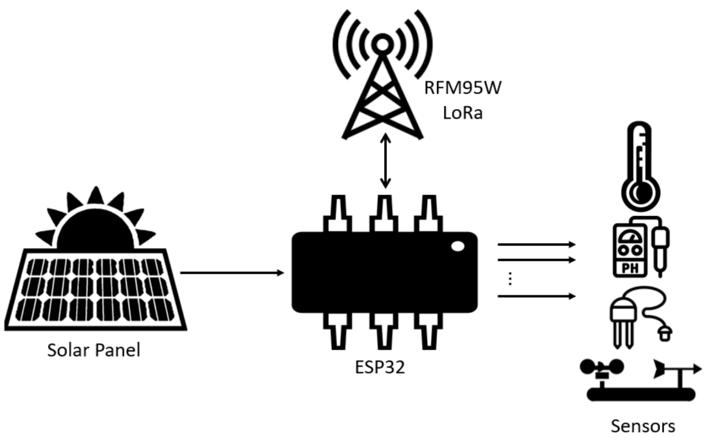


FIGURE 3.4. Sensor Node

The size of the battery and solar panel are described in Section 5.1.3, as far as the calculation of the solar panels dimensions according to the node consumption's.

Regarding to the data collection, several sensors were used, such as:

- SI7021, Figure B1, a temperature & humidity sensor, capable of collect temperature and moisture values with high precision (± 3%RH, ± 0,4°C), in a range between -40°C and 125°C in terms of temperature and 0 to 100% in terms of humidity values [**41**].
- DS18B20, Figure B2, a waterproof soil temperature sensor, capable of collect data between -55°C and 125°C with an accuracy of 0.5°C for -10°C to 85°C interval, easy to implement, with a one wire connection [**42**].
- Analog Capacity Soil Moisture, Figure B3, a humidity waterproof sensor which provides the capacity of collecting data with high precision. This collection is made by capacitive sensing [**43**].

These sensors are connected to the Analog to Digital Converter (ADC) interface of the ESP32, making possible to control them and collect the values and send them to the Aggregation Node. To control the sensors, it was necessary to use the respective libraries, which were: DallasTemperature.h to control DS18B20, Adafruit's library to control Si7021 and OneWire.h to control Analog Capacity Soil Moisture.

After the data was collected, a message was created that contains the node id, the sensor nickname that collected each value and the respective value, as explained in the previous topic. Finally, this message is sent via LoRa to the Aggregation Node.

In order to keep the data of the field under analysis updated, a collection is made every 10 minutes. Between each collection, and in order to save the battery that powers the node, the circuit was placed in Deep Sleep mode where the energy consumption was reduced to 100 $\mu$A. At the end of each Deep Sleep time the node was woken up and runs all the code again, collecting and sending new data.

### 3.1.4. Actuator Node

The Actuator Node was in charge of acting according to the collected data by the Sensor Node. This node consists in a microcontroller, ESP32, a LoRa radio module, RFM95W, the weather station and the irrigation pumps. As explained before the RFM95W will be responsible for exchanging messages between the node and the Aggregation Node, receiving the irrigation instructions and sending the current status of each pump.

ESP32 is responsible for controlling all modules of the node, it will control the received data, process them and finally turn the pumps *ON* or *OFF*. Besides the spoken sensors in the previous topic, the microcontroller will also be connected to a weather station which is composed of three sensors which needs to be constantly collecting data, being this reason the station was installed in this node, and not on a Sensor Node. The SEN 0186 [**44**], a Weather Station which provides the capability of collect environmental data on the field, is equipped with an anemometer, used to measure wind speed, a wind vane, for collection of wind direction and a rain bucket, in order to collect precipitation values.

Regarding the node power consumption, and since the irrigation pumps used work at 12V, it was not possible to power the node by batteries or solar panels, similar to the Sensor Node. Therefore, the node will have to be powered by an electrical socket with a transformer that converts the 220V of the local electric current to 12V needed. Also, a current converter, the LM2596, was used to converts the 12V coming from the transformer to 5V, in order to power the ESP32 and the whole circuit.

Since ESP32 does not have enough power to supply the water pumps the Panasonic AQY212EHAT Solid State Relays (SSR) was used, making possible, through a simple electronic circuit and HIGH and LOW signals from ESP32, to switch *ON* or *OFF* the 12V supply to the water pumps, switching there status. The Actuator Node constitution is shown in Figure 3.5.

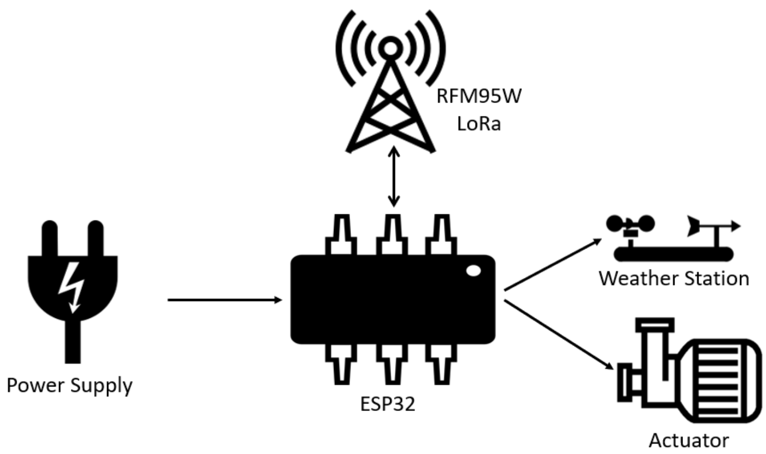


Figure 3.5. Actuator Node

As far as data collection from the weather station is concerned, even though it must always be on, values are only collected every 10 minutes, like the Sensor Node. This need is due to the fact that the station is always updating the values of wind speed, wind direction and rainfall, making an average of the last 5 minutes and returning it. After every 10 minutes the microcontroller collects this data and sends it to the Aggregation Node, following the structure explained in the Aggregation Node topic for the exchange of messages containing the data of the Sensor Nodes.

When it comes to water pumps control, ESP32 is always listening to messages from the Aggregation Node. Whenever a message arrives containing the id of the destination node, the nickname of the pump that will change its state, the state to which this pump is to be changed and, if the message indicates to turn it *ON*, it indicates how long it will be *ON*. The ESP32 puts the corresponding port in the desired mode (HIGH/LOW) and starts counting the time that it will be in HIGH mode, if that was the indication. When turned *ON*, the water pumps starts working, having a water flow of 5.83 l/min.

## 3.2. Data Analysis

In order for the system to work as intended, it was necessary to create a central point where all the information gathered converges, the necessary calculations were made and the necessary messages were sent. This central point is the server and for which several Python scripts have been developed in order to guarantee all the logic of the system.

Once again, some previous work was already done by [**13**] and [**14**], but on opposite of the hardware part, which was the focus point of those works, little to none of the server side can be used on our solution. This decision is mainly due to:

- Outdated technologies or techniques, mainly in the creation of an API using PHP with multiples files, instead of single RESTful application;
- Dashboard developed only for Android;
- Little to none use of Machine Learning to improve data analysis and decision making.

So, in terms of server-side software, all the work presented was built from scratch, without the use of the previously done work.

The server side part of the proposed system is composed by five key modules: database, and all the scripts associated to collected the data; Irrigation Algorithms, responsible for calculating the real amount of water needed based on sensor data; Machine Learning algorithms, to understand the best irrigation hours based on sensor data; RESTful API, to feed data to the dashboard; and the Dashboard, to allow the user to control the entire system. The developed modules are described in detail in the sections below.

### 3.2.1. Database & Support Scripts

As far as the database is concerned, it was created and controlled through MySQL, a database management system that uses the SQL language to insert, access and manage the content stored in the database [**45**]. In order to structure the whole system several tables were created, each one being responsible for storing the data of each element of the system. The structure and the tables created can be seen in Figure B4.

Besides the place where the data is stored, it was also necessary to develop a set of scrips that work as intermediaries between the data received through the MQTT and the database. For this, a Python script was developed, *mqttBD.py*, being this script responsible for subscribing, receiving and treat messages from two MQTT topics.

To the first topic, *"thesis/data"*, comes the messages from the Sensor Node with the last collected data. Whenever a new message is published in this topic, it is analysed to understand from which node id the message came from, to make sure that the message comes from one Sensor Node, and then it is analysed to understand which data corresponds to each sensor. After this analysis an INSERT SQL code is created and later sent to the database to add a new entry.

Regarding the second subscribed topic, *"thesis/BD"*, it refers to the status of the actuators. Whenever a new message is published, the message is analysed and an UPDATE SQL code is created and later sent to the database.

Besides the information from the sensors and actuators coming from the field, the database also stores weather forecast data from Instituto Português do Mar e da Atmosfera (IPMA) [**46**], the entity responsible for Portuguese weather forecasting. For this a Python script was developed, *ipma.py*, which, through an Application Programming Interface (API) provided by IPMA, collects the daily weather forecasts of the location where the system was installed, as well as the information for the next three days. This script is executed automatically, every day at midnight, in order to ensure the forecasts for each day are updated. This data will later be used to make calculations that will return the irrigation time to be administrated for each zone.

### 3.2.2. Automation & Knowledge creation scripts

As the main goal of this work is to create an autonomous way to improve the efficiency and sustainability of irrigation in agricultural sites, based on an IoT and WSN system, a way to analyse the sensor data collected and transform those information into knowledge is needed, mainly to control the real amount of water needed or the best time of day to irrigate. For that, two different approaches were done, one only based on calculations using the sensor data, to discover the optimal irrigation time, and a second one based on a machine learning approach, using the sensor data to predict the best irrigation hour and them calculate the irrigation time, with the previous approach.

**3.2.2.1. *Irrigation Algorithm.*** Taking into account the main objective of the developed system, water saving, formulas were studied and implemented in order to calculate how much water is necessary to administrate to a specific irrigating zone based on the data collected by the sensors on that zone.

These formulas were developed by [**47**] taking into account the use of soil moisture and air temperature and humidity sensors and considering the type of crops, the type of valves and tubing used, the distance between these same valves and the number of irrigation's in one day. Considering all these parameters, and using Equation 3.1, it becomes possible to calculate the amount of water for each irrigating zone.

$$\mathrm{T} = \frac{\mathrm{A} \times (\mathrm{K_c} + \mathrm{ET}) \times 60}{\mathrm{F} \times \mathrm{N} \times 1000} \tag{3.1}$$

In the Equation 3.1, the result, $T$, represent the irrigation time in minutes, $K_c$ is the crop coefficient, according to Table A1, $ET$ is the evapotranspiration in $mm/day$, resulting from Equation 3.3, $F$ is the outgoing flow of water in $m^3/h$, $N$ is the number of valves, $P$ is the number of irrigation periods and $A$ is the garden area in $m^2$, this last is given by Equation 3.2.

$$A = [(0.5 \times N) - 1] \times D^2 \tag{3.2}$$

In Equation 3.2, $D$ represents the distance between valves.

$$\mathrm{ET} = 0.0023 \times (\mathrm{T_{med}} + 17.78)\mathrm{R_o} \times (\mathrm{T_{max}} - \mathrm{T_{min}})^{0.5} \tag{3.3}$$

In Equation 3.3 it is possible to see the simplified Hargreaves formula. This one will give the evapotranspiration ($ET$) in $mm/day$, where $T_{med}$ is the average temperature, $T_{max}$ the maximum temperature, $T_{min}$ is the minimum temperature and $R_0$ is the incident extra-terrestrial solar radiation in $mm/day$, according to Table A2.

To account for the sensor values from soil moisture and air humidity, in order to get a optimized time for irrigation the author of [**47**] formulated Equation 3.4.

$$\mathrm{T_{opt}} = \mathrm{TI_{soil}} \times 0.7 + \mathrm{TI_{hum}} \times 0.3 + \mathrm{T} \times 0.1 \tag{3.4}$$

In this, $TI_x$ is the time needed to reach 100% of the variable $x$, soil moisture (soil) and air humidity (hum), respectively, given by Equation 3.5, where $I_x$ is the last value collected for sensor $x$.

$$\mathrm{TI_x} = \frac{\mathrm{T} \times (100 - \mathrm{I_x})}{100} \tag{3.5}$$

Besides this approach gave a 34% improvement when used in a grass irrigation system in [**47**], as the author state it can be still improved using more sensor data to optimized even further the formula. For that, and since our proposal adds a weather station to the system, we can improve the formula using the values for rain and wind.

As such, the first thing to add is the effect of rain in irrigation. Equation 3.1 shows that for irrigation the crop coefficient, $K_c$, summed with the evapotranspiration value, $ET$, represent the mm/day of water needed for a healthy crop. But, when rain falls in that area that amount of water might already been put into the crops, and using the rain gauge from the weather station it is possible to know exactly the amount of rain in the last 24 hours, so that value can be subtracted from the previous one, indicating exactly the amount of water the crops still need for that day. So it is possible to add the last amount of rain for the last 24 hours, $TI_{24rain}$, to Equation 3.1 and optimize that formula.

$$\mathrm{T} = \frac{\mathrm{A} \times (\mathrm{K_c} + \mathrm{ET} - \mathrm{TI_{24rain}}) \times 60}{\mathrm{F} \times \mathrm{N} \times 1000} \tag{3.6}$$

Finally, and as the system must react to real time changes in the environment, the following methodology was used to stop the irrigation due to rain or strong winds, as if it is raining there is no need to be irrigating and if strong winds occurs, they can blow away the water, so irrigation must stop and be scheduled for another time. For that the threshold of 1.0mm of rain in the last 5 minutes or wind speeds over 35 m/s were set to indicate whether irrigation can start or if it is scheduled for another time.

Based on the formulas presented, a Python script was developed, *WaterringAlgorithm.py*, that calculates the irrigation time needed for a specific zone, based on all the sensor data collected by the system. Each irrigation zone has, as parameter, the time of day destined to its irrigation, and as such, the script will check, every minute, if there is any zone to be irrigated at that moment and, if this is the case, calculates the irrigation time to manage the zone in question and sends a message via MQTT to the Aggregation Node with the calculated information, following the structure presented in Section 3.1.2.1.

**3.2.2.2. *Machine Learning.*** As explained in the previous topic, it became possible to estimate the time of irrigation needed to be administered to an irrigation zone, however the time of day when this water administration was done had to be indicated manually by the user, leading to irrigation being administered at a less favourable time of day leading to water wastage. To make the system more complete, by removing the need for manual irrigation time input, a machine learning algorithm was developed to predict the most appropriate time of day for water administration.

The developed script implements a machine learning algorithm that, when receiving as input the current time, air temperature, air humidity, wind speed and direction, soil humidity and whether or not it is favourable for watering, the script returns the most suitable time for irrigate the zone under study.

The methodology for this approach, as shown in Figure 3.6, is to receive the data in real time from the system, pre-process them and put them through the machine learning algorithm in order to understand, based on a previous trained model and dataset, if that hour is suitable for irrigation and if not, which will be the best hour. From time to time, the model is retrained using a new dataset that includes all the new data that was collected since the last train.

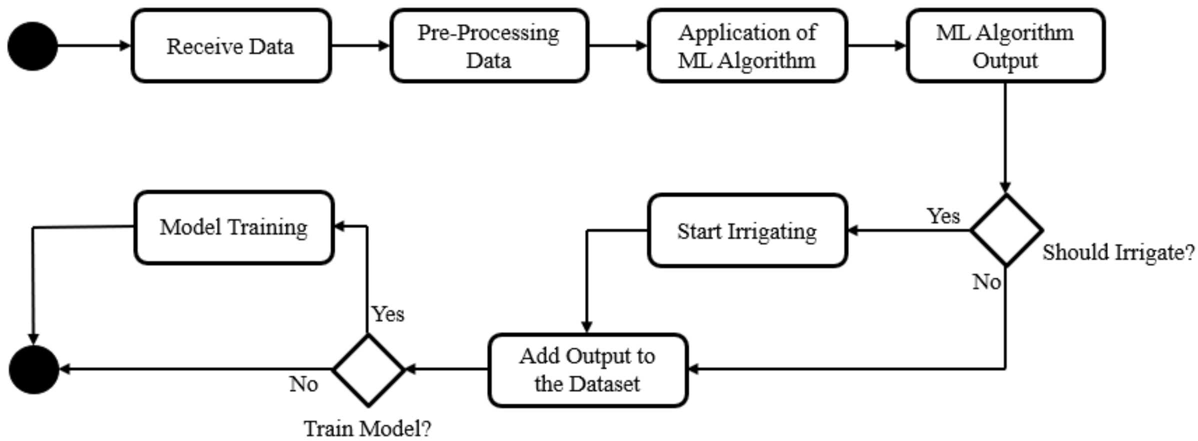


FIGURE 3.6. Machine Learning Methodology

The script, *ML.py*, was implemented using Python and is deployed every hour, to predict the best time to irrigate each zone taking into account its current conditions.

The study of the best model, the dataset creation and the data pre-processing will be explained in detail in Chapter 4.

### 3.2.3. Application Programming Interface

As previously mentioned, a dashboard was developed to consult and control the developed system. In order to this dashboard have access to all the system's information, it was necessary to give it access to the database in order to make selects, updates and inserts. To provide these features a RESTful API was developed. Representational State Transfer (REST) and Application Programming Interface (API), these two technologies were used to develop a set of programming routines and standards in order to give to software applications access to web-based platforms, in this case to the database. In order to develop this solution, it was created a Python script where it was defined a set of parameters and restriction to the made requests by the application. These requests were sent through HTTP packages, as this protocol offers a set of operations (GET, POST, HEAD, PUT, DELETE, etc.) to use in order to correspond to all the situations, with the developed solution only using GET and POST.

Each time the application needed to access the database a URL path is created, where it defines the target of each request, for example *"https://smart-farm.pt/5001/actuator/2"*, meaning that is requested the data stored on the database which concerns to the actuator with the id number 2. In this example the request used the GET operation, since it was pretended to get information from the database. The POST operation it was used when it was pretended to make an update or an insert to the database.

The developed Python script is running on the server and waiting to income requests, that are analysed and, if the structure matched to the restrictions developed, the pretended tasks and responses are performed. In the case of the income request was a GET method, it was created a JSON array where all the information requested was divided and prepared to be sent to the requester.

### 3.2.4. Dashboard

In order to analyse the collected data and perform the needed tasks a mobile application was developed, to offer the user the possibility to perform tasks without the need of be in the field location. The developed application is called “Smart-Farm” and its logo is shown in Figure 3.7.


FIGURE 3.7. Application Logo

This mobile application was developed in Swift, using his most recent version Swift 5, through his official developing environment XCode. Either the used tool and the used language were produced and own by Apple and the operation system targeted was iOS.

As said before, the developed application has the functionality of showing the collected data and the possibility to preform task. For this, it was necessary to give it the ability to connect to the developed API and the tools to use MQTT in order to perform tasks into the field, for example turning on one actuator.

In order to give the right logic to the application it was necessary to create multiple views, Figure 3.8, being each one panel shown in the application and responsible for show specific information or text panels.

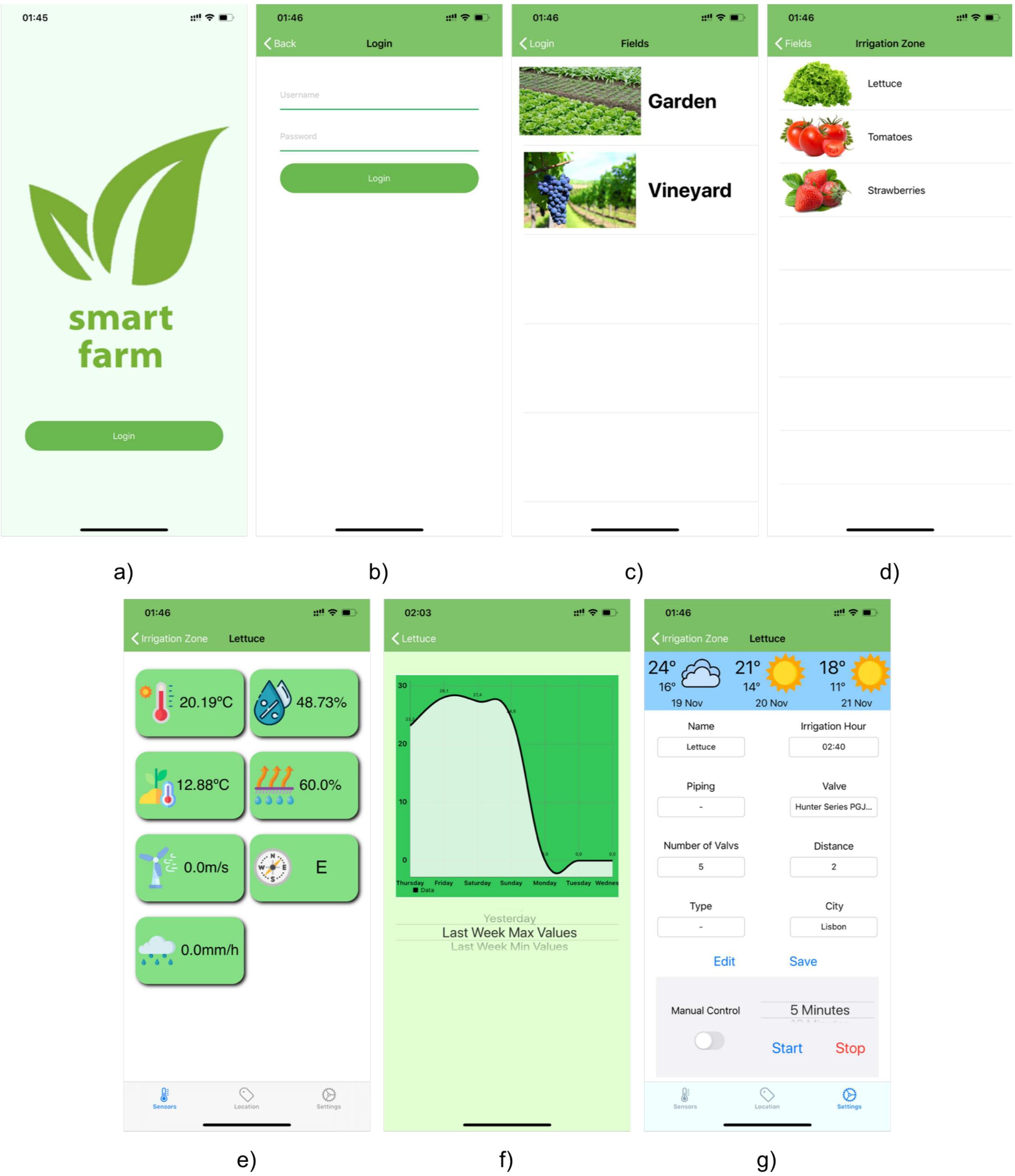


FIGURE 3.8. Dashboard Application views - a) Start View; b) Login View; c) Fields View; d) Irrigation Zones View; e) Sensors Values View; f) Sensor History View; g) Irrigation Configuration View

The application starts by showing a home screen view, Figure 3.8 a), with a start button. When this button is pressed the view is changed and takes the user to the authentication view, Figure 3.8 b), where he is prompt to input is username and password. If the authentication is correct, the dashboard presents to the user all the systems that he as connected to his account, each one representing a different agricultural field, Figure 3.8 c). Here the user can select an individual field and is forward to a new screen showing all the sections associated to that agricultural field, Figure 3.8 d), each representing an individual irrigation zone.

Just like the fields view, by choosing and clicking in one of the irrigation zones, the application leads the user to the data view, Figure 3.8 e), where the last collected values from the sensors installed in the selected zone are shown. In order to analyse the variation of the collected values from each sensor, by clicking on the desired sensor, a new view, Figure 3.8 f), presents a chart with all the collected values from that sensors and the corresponding timestamp. To help filtering the data, the user can choose the timestamp we wants to be displayed in the graph. The final view, Figure 3.8 g), allows the user the possibility to perform tasks on the field, mainly in order to change the irrigation zone settings, consult the local weather forecast, turn *ON* or *OFF* the water pump of each irrigation zone and turn *ON* or *OFF* the auto irrigation mode.

CHAPTER 4

# Machine Learning Training

With the constant evolution of technology and the appearance of new solutions that, when combined, manage to achieve sustainability, the exploration of these systems is increasingly a path to take. This way a study and development of machine learning algorithms was made with the aim of predict the most suitable time of day for water administration to an agricultural field.

Machine Learning (ML) is a technology which as the capacity of learn and improve through his own experience. These improve capacities are possible due to the access of a huge amount of data previously granted to the system and which is constantly updated with new data, as the algorithm is exposed to new situations and must give an answer for that. One of the great characteristics of ML is that all this constant and automatic learning is done without any human interaction.

In order to study the most suitable classification algorithm, the following machine learning algorithms were considered: Random Forest (RF), Neural Networks (NN), Extreme Gradient Boosting (XGBoost), Decision Trees (DT) and Support Vector Machine (SVM). In addition, and in order to be able to study correctly the algorithms already mentioned, a methodology was created, leading to the best possible optimization.

## 4.1. Methodology

The followed methodology for the development and improvement of the chosen ML algorithms was divided into 4 phases (Figure 4.1). In a first phase a dataset was built through the collection of data by the presented system. After the dataset was completed, all values were analysed and each entry classified by three additional features, as: favourable or unfavourable for water administration, whether or not this same water needs to be administered and if the land has already been watered ("Is_favourable", "Need_Irrigation" and "Had_Irrigation" respectively).

In a second phase, with the completed dataset, and before testing the chosen algorithms, a test on the importance of each feature of the dataset was made in order to understand which were the most important and which should not be considered at the time of training the algorithms in order to optimize the dataset and leading to the elimination of noise.

The third phase of the methodology consisted in training the algorithms under study using the parameters of each one with default values, to check the accuracy of each one.

In the fourth and last phase of the followed methodology, the best algorithms, from the precious phase, were exhaustively tested in order to understand what would achieve better accuracy results. In these tests a hyper parametrization tuning was done to each algorithm, in order to understand the best scenario possible. For this, a method provided by scikit-learn called RandomizedSearchCV was used, which performs the fit and training of the algorithm under study, calculating which parameters are best suited to it [**48**].

The final algorithm was then adapted to be running on the server, where the database was hosted and where the values arrived in real time. Thus, the algorithm receives the collected values and calculates the best time for water administration. When crossing the resulting values of the algorithm developed with the algorithms previously spoken and already developed, that calculate the amount of water needed to be administrated to the field under analysis, it will lead to a better management of this same terrain and obtaining very high water saving values, being this the main objective.

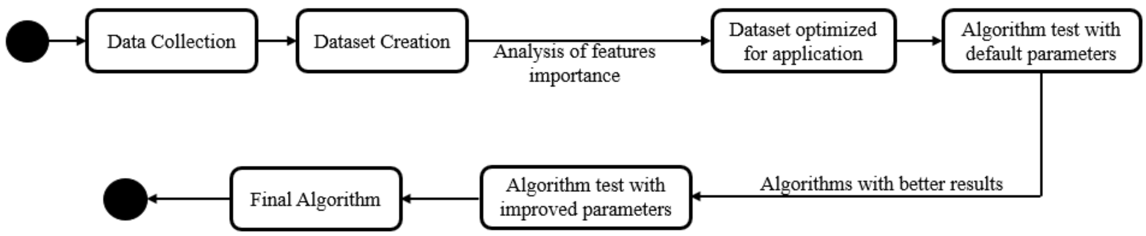


FIGURE 4.1. Machine Learning Methodology

### 4.2. Dataset Creation

In order to develop the correct algorithm that could predict the best hour of day for water administration, it was necessary to compose a correct and useful dataset.

With the aim of collecting the necessary amount of data needed to create the pretended dataset, a prototype of our system was used to collect the necessary data, for air temperature and humidity, soil temperature and moisture, rain and wind speed and direction, in a small urban farm.

After the conclusion of the collection phase, the data was compared and complemented, when necessary, with values provided by IPMA, being able to confirm the accuracy of some of the collected data since some of this were available on IPMA API.

Besides the dataset features provided by the implementation of the presented system, all entries were individually pre-processed and analysed, with extra features being added to each one, making the dataset richer and more substantiated in order to make it easier the process of training the algorithms under study and achieving better results. These new features are: "Is_Favourable", "Need_Irrigation", "Had_Irrigation" and "Suggested_Hour".

Regarding to "Is_Favourable" feature, this one refers to the weather and field conditions, indicating whether they point to favourable or unfavourable conditions for sustainable irrigation. This feature was calculated through some conditions, such as: the wind speed under 36 m/s; the wind speed under 25 m/s and the wind direction between 180 and 359 degrees; the rainfall values under 1 mm/h; the air temperature under 30°C; the hours not in between 10:00 and 16:00. If all these conditions were verified, was considered favourable for irrigate.

The "Need_Irrigation" feature, was calculated through two conditions, as the feature presented above, being these as follow: the soil moisture under 60%RH with the air temperature over 25°C or the air humidity under 40%RH; or the soil moisture under 45%RH, disregarding weather conditions. If one of this condition is checked, the field needs to irrigate.

For the "Had_Irrigation" feature, it only indicates if the field was irrigated in the day in question.

Finally, regarding to the "Suggested_Hour" and since this would be the features that the Machine Learning had to predict, this feature was added manually to each entry, ensuring that the suggested hour was adapted to each weather and field condition.

After conclusion, the dataset used has 105217 entries and had the features presented in Table 4.1.

### 4.3. Model Analysis & Remarks

In order to put into practice the methodology already described, it was necessary to precede to the training and subsequent testing of the various algorithms under study to understand what best suits the intended application. To understand if the algorithms have a good applicability when receiving real values, the dataset was divided in two

Table 4.1. Dataset Properties

| Feature | Description |
|---|---|
| Year | Timestamp Year |
| Month | Timestamp Month |
| Day | Timestamp Day |
| Hour | Timestamp Hour |
| Temperature | Air Temperature [°C] |
| Relative_Humidity | Air Humidity [%] |
| Total_Precipitation_Low | Precipitation [mm/H] |
| Wind_Speed | Wind Speed [km/h] |
| Wind_Direction | Wind Direction [°] |
| Soil_Humidity | Soil Moisture [%] |
| Had_irrigation | Field irrigated [0/1] |
| Need_Irrigation | Field needs irrigation [0/1] |
| Is_Favorable | Conditions favorable for irrigation [0/1] |
| Suggested_Hour | Suggested irrigation hour |

parts. A first part that represents 70% of it, that will be used to train the algorithm, and a second part, which represents 30% of the dataset, which was used after the training of the algorithm to understand its accuracy.

As explained in the methodology, before the study of the chosen algorithms, a test was performed in order to understand which were the most important parameters of the dataset. The results obtained can be seen in Figure 4.2.

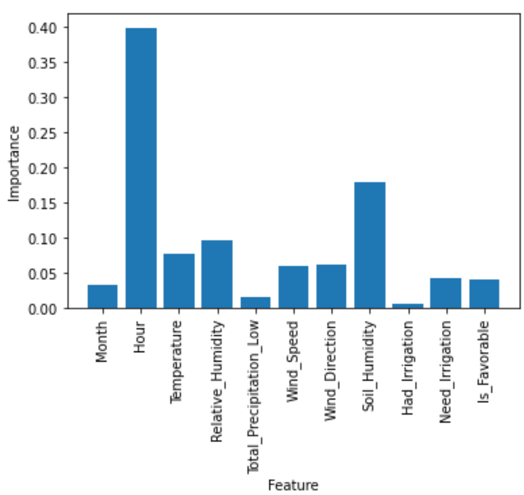


Figure 4.2. Initial Features Importance

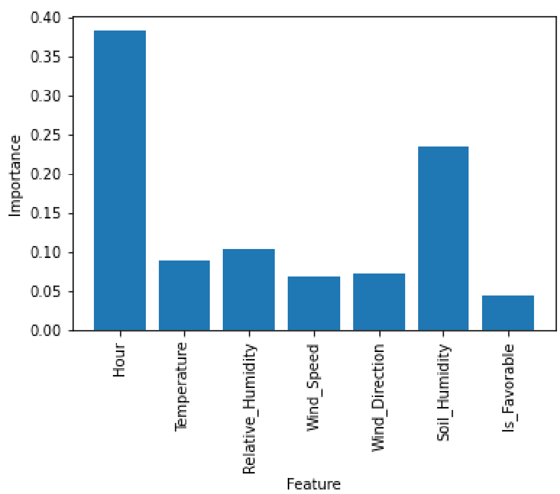


Figure 4.3. Final Features Importance

As can be seen, the features "Day", "Need_Irrigation", "Month", "Had_Irrigation" and "Total_Precipitation_Low" have a low importance for the training and later result of the algorithms, so they were discarded. This discarding also results in a shorter time in the training of the algorithms and no high variation was observed in the results obtained after the discarding. The final results were obtained only using the features that can be seen in Figure 4.3.

Regarding to the third phase of the methodology, the five algorithms under study were trained using only their defaults parameters. For this scikit-learn, an open source Machine Learning library developed for python implementation [**49**], was used. For the implementation of XGBoost, a library made available by this same algorithm was used. It implements machine learning algorithms under the Gradient Boosting framework [**50**]. Table A3 shows the default parameters used for each of algorithms.

The accuracy results obtained for each model are presented in Table 4.2. As can be seen, the accuracy values of each algorithm trained were quite varied, allowing to conclude that each algorithm has it purpose and that is always important to study which better fits to the used dataset and pretender output.

Table 4.2. Default Classification Results

| **Algorithm** | **Accuracy**[%] |
|---|---|
| XGBoost | 86.57 |
| Random Forest | 84.77 |
| Neural Network | 81.71 |
| Decision Tree | 79.92 |
| SVM | 31.38 |

It is possible to notice that the SVM does not fit in the proposed goal, having a low accuracy value and, even with the variation of some parameters, it would not be possible to achieve acceptable values. As for the other algorithms, although higher when compared to SVM, it is possible to conclude that only two stand out even before any optimization. As such, XGBoost and RF will be further evaluated, since DT and NN, even after optimized, will not be able to reach better accuracy.

Moving forward to the next training phase, using only XGBoost and RF, and based on the parameters shown in Table A3 and the documentation for each of these algorithms, the hyperparametrization tuning was made only for the most important parameters, mainly those who have a numerical value. Table 4.3 shows the tuning options for each of the selected algorithms.

Table 4.3. Default Configuration Parameters

| **Algorithm** | **Parameter Configuration** |
|---|---|
| XGBoost | $max\_depth = randint(1, 500), n_estimators = randint(1, 500),$ $subsample = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1],$ $tree\_method = ['auto','exact','approx']$ |
| Random Forest | $n\_estimators = randint(1, 500), criterion = ['gini','entropy'],$ $max\_depth = randint(1, 500), min\_samples\_split = randint(1, 100),$ $max\_features = ['auto','sqrt','log2'], bootstrap = [True, False]$ |

The results obtained with the hyperparametrization tuning, including the best parameters settings and accuracy, can be observed in Table 4.4. Through the analysis of the results, after the optimization of the various parameters of the two algorithms under study it is then possible to conclude that by choosing the best parameters, instead of the default configuration, it is possible to improve the accuracy of the models. Although is not a huge improvement, 1% for XGBoost and only 0.1% for RF, this improvement can lead to the saving of a huge amount of water.

Table 4.4. Optimized Configuration Parameters & Results

| **Algorithm** | **Parameter Configuration** | **Accuracy[%]** |
|---|---|---|
| XGBoost | $max\_depth = 48,$ $n_estimators = 324,$ $subsample = 0.5, tre\_method = auto$ | 87.73 |
| Random Forest | $n\_estimators = 212,$ $criterion = gini, max\_depth = 196,$ $min\_samples\_split = 10,$ $max\_features =' sqrt',$ $bootstrap = True$ | 84.74 |

Since XGBoost was the one that obtained the best accuracy for the situation and the used dataset, this last will be the one used for the solution.

CHAPTER 5

# System Implementation

In Chapter 3 it was described the pretended architecture for the proposed system. The proposed architecture is composed by multiple components due to its capabilities of collecting, receiving and sending data. For this, and as explained in Chapter 3, it was necessary to use multiple solutions and technologies. In this chapter will be explained the system setup where laboratory and field tests were performed to investigate the feasibility of the chosen technologies, the assembly of the complete system and its setup into the field.

## 5.1. System Setup

Regarding to the performed laboratory tests, these ones consisted in the implementation and test of each part that compose the system in order to understand if it meets the intended requirements, so that adjustments could be made before the final implementation.

### 5.1.1. Communications

One of the key features of the proposed system was the communication capabilities, being necessary to test the communication established between each node and the communication between the Aggregation Node and the server. Concerning to this topic, it was followed the same test made by [**13**] and [**14**] in their research in order to ensure the correct assembly of the node, since the LoRa solution was already evaluated by the mentioned studies. When it comes to the communication between the Aggregation Node and the server, it was followed the same test made by [**13**] and [**14**] in order to ensure the correct performance of MQTT, however for the presented system, it was used a different technology to establish connection to the internet, NB-IoT, and this one had to be tested, ensuring the correct operation.

Regarding the communication between nodes, it was tested the viability of the LoRa protocol and modules. For this, it was created a simulation of two nodes, Figure 5.1, with the same hardware which will be used in the final implementation. In a first step the nodes was placed near each other and a message was sent every 30 seconds. At a later stage, it was increased the distance between them, simulating the distance that must be ensured when installed in the fields.

The first described test had a 100% performance, since every sent message was correctly received. Regarding to a second moment of the test, when the distance was increased, it was noticed that the communication was still able to be conducted, however when an obstacle is present, in this case a wall, about 20% of the sent messages did not reach the intended target, since no ACK was received.

Although this situation occurred, every time a message was lost, our developed acknowledgment system, that ensures the overcome of messages lost, retried the delivery until and ACK is received, or until five attempts were failed. With this, after the 30 minutes of test, no messages were lost, being confirmed the capabilities of the chosen modules to ensure the communications between nodes in the field.

In a second stage, with the objective to create a simple WSN, it was added a sensor to one of the created nodes, simulating a Sensor Node and making the other one the Aggregation Node. In this test was possible to ensure the capability of the system to correctly collect data and send it through LoRa.

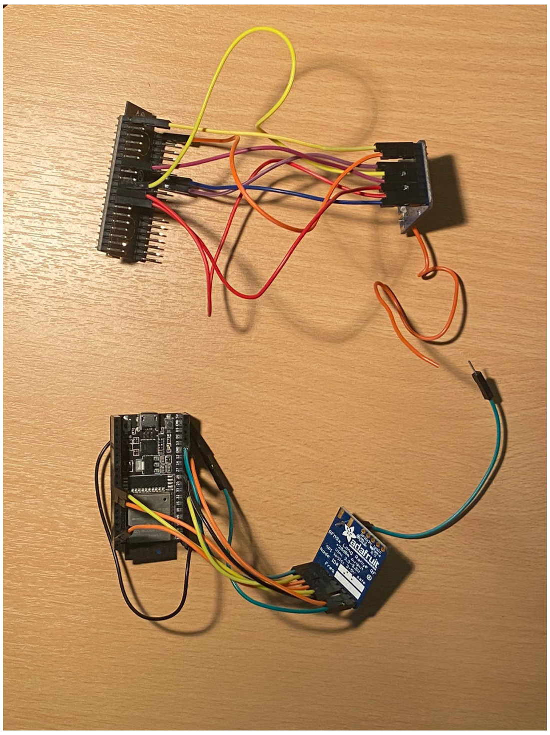

FIGURE 5.1. LoRa Test Scenario

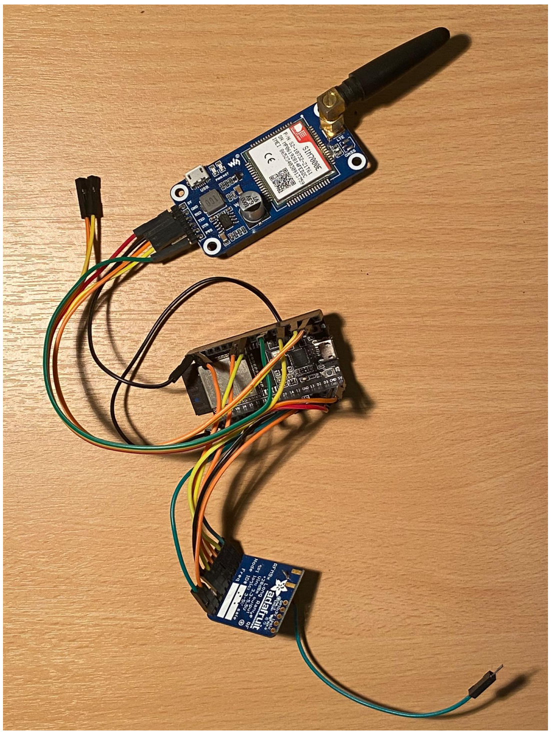

FIGURE 5.2. NB-IoT Test Scenario

For the communication with the server, the performed tests followed two phases. In a first phase, since the laboratory tests was made in a controlled environment with Wi-Fi coverage, it was used the ESP32 Wi-Fi capabilities in order to establish an internet connection and allowing the usage of MQTT. For this test, the Aggregation Node worked as publisher and it was used a WebSocket to play the role of subscriber. Each sent message had an interval of one minute. The performed tests went as expected and all the published message was received correctly.

Regarding to the second phase, it consisted in the replacement of the Wi-Fi technology by NB-IoT. For this, the NB-IoT module was installed in the Aggregation Node, Figure 5.2, and programmed in order to being able to establish an internet connection through cellular network. The rest of the procedure was the same as the first stage and it was possible to ensure the correct message sending and the respective reception.

### 5.1.2. Data Collection in Real Environment

Since the proposed system has a high component of data collection, it was performed a data collection test in a real environment. The goal of this test was to unsure the correct data collection by the system, so that when the actuators were installed, the collection part works correctly, and to understand how the different irrigations zones react when exposed to the same situations. To guarantee the correct outcome is seen, this test was performed for three weeks in a controlled domestic urban farm. For this the Aggregation Node, the Actuator Node and one Sensor Node were installed. Although the goal was to test the data collection, the Aggregation Node was used to simulate the installation of a complete system, being able to also test in a real scenario the transmission of data to the server. The Actuator Node was also installed since, as already described, it is the node responsible for managing the weather station, that collects important values for the system, as seen in Figure 5.3.

The test focus on monitoring two irrigation zones, and for that, the Sensor Node was installed using two soil moisture and two soil temperature sensors, and the temperature and humidity sensor. So, for this test data of soil moisture, soil temperature, air temperature, air humidity, wind direction, wind speed and rainfall was collected. Figure 5.4 shows the field installation where the two irrigation zones under study were divided.

**5.1.2.1.** ***Remarks.*** The results obtained from the test carried out shown a variety of results from which some conclusions could be drawn. Since the performed test collected

FIGURE 5.3. Weather Station

FIGURE 5.4. Collection Test

values of air and soil temperature, it was possible to observe the differences between these two.

In Figure 5.5, it was possible to observe the air and soil temperature variation over one day. The day under review shown a maximum air temperature of 28°C registered at 14:20, and a minimum value of 18°C registered at 4:33 of that day. By moving on to the analysis of the soil temperature collected values by the two installed sensors in

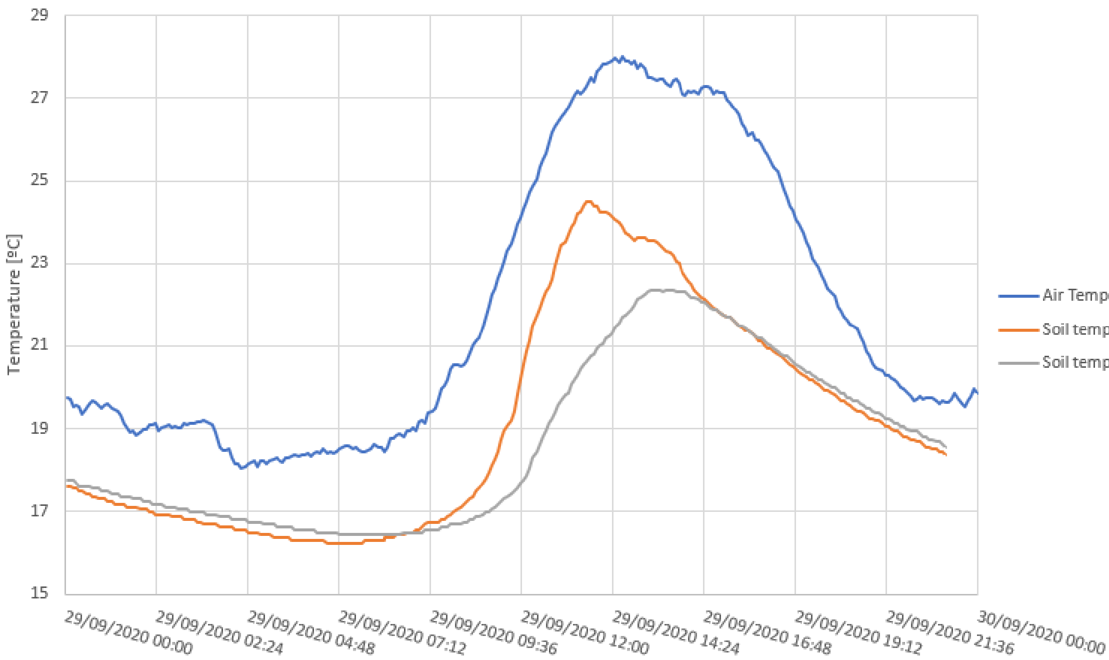


Figure 5.5. Air and Soil Temperature Comparison

both irrigation zones, it was possible to conclude that there was a temperature variation between the two tested zones. Zone 1 had a maximum temperature of 24.5°C at 14:10 and zone 2 had his maximum registered temperature at 16:07 with 22.5°C. Through a simple analysis of these three lines it was possible to conclude that the temperature variation on the air is much more pronounced when compared to the variations observed in soil temperature records, where these are more gradually without any abrupt variation.

Regarding to the difference between the collected data from the two irrigation zones, although the distance between them is not very high and the conditions are the same, it is possible to observe a difference in the variation of their values, where it is possible to conclude that irrigation zone number two does not reach such high temperature values when compared to zone number one. This will have an impact on the need for irrigation, contributing to different needs for the same crop and location.

Another analysis which was possible to made through the collected data was the relation between the rainfall and the moisture values in the two irrigation zones in analysis. In Figure 5.6 is possible to observe the rainfall values variation during a rainy morning and the moisture values collected for the same day by the two soil moisture sensors installed.

Through the analysis of the Figure 5.6 is possible to realize that between 8:48 and 11:57 it was raining, with the amount of rainfall varying through time, with a maximum collected of 2.7 mm/h. Although the rainfall values collected were not very high and, on that day, only rained in the morning, it was possible to draw some conclusions through the analysis

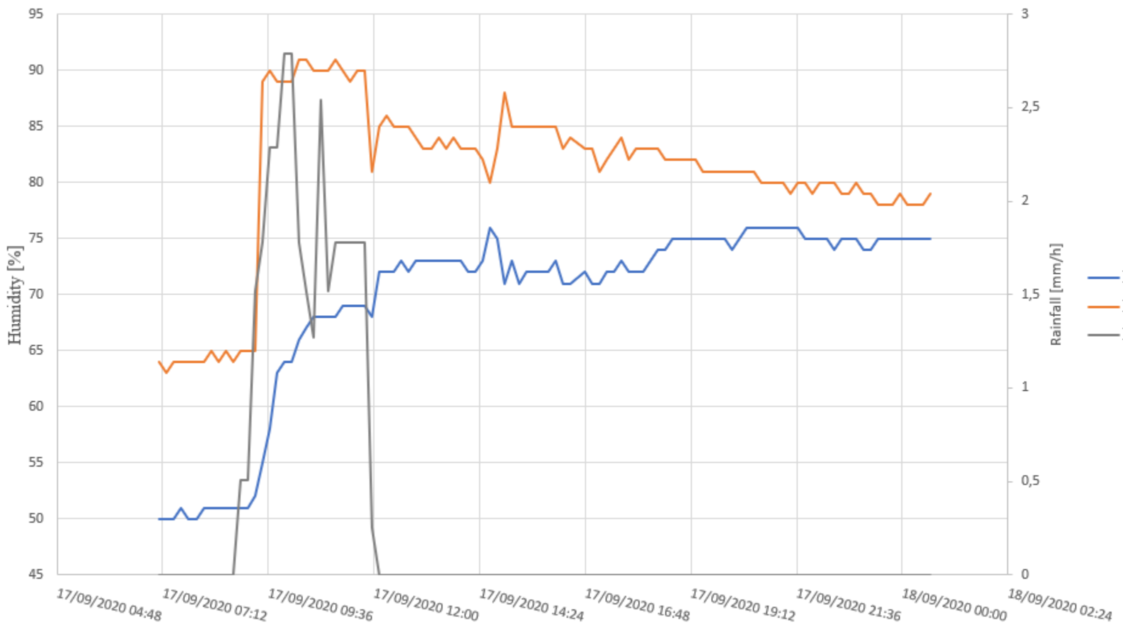


FIGURE 5.6. Soil Moisture Variation During Rain

of the impact of this rain values on the variation of the collected soil moisture values. As mentioned before, two irrigation zones were monitored being one of the principal reasons for this the inquire of the data difference between two zones relatively close to each other. Through the analysis of the graphs presented before, it was possible to observe the increase of soil moisture values from about 30 minutes after it started raining. However, the collected values of each zone were quite different, showing that the soil moisture values of zone one were always lower than in zone two and, when started to rain, the values of zone two increased in an abrupt way and, in contrast, the values of zone one have gradually increased. After the rain stops the values stayed stable during the rest of the day.

Since one of the installed sensors was the wind direction sensors, installed in the weather station, it was possible to collect the values of wind direction of the location where the test was performed. Figure 5.7 shows the data collected from this sensor during the entire test. As can be seen, is possible to conclude that the wind direction of that location is, for most parts, felt for the East.

### 5.1.3. Power Consumption

The developed system has several key points that make it able to offer the ability to perform all the pretended features. One of these key points is the power supply of each node of the system. As explained before, the part of the system developed to be installed in the field is composed by three types of nodes: Sensor Node, Aggregation and Actuator

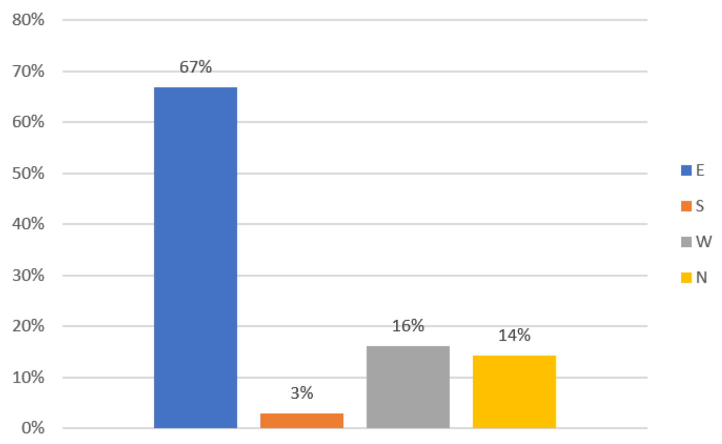


FIGURE 5.7. Wind Direction

Node. The last two, regarding to their tasks, had to be always *ON* since they need to be collecting, receiving or sending data, being these connected via electrical current and powered by an 5V converter. In the other hand, the Sensor Node only needs to be *ON* when is collecting and sending data to the Aggregation Node, being in Deep Sleep mode between collection times. This situation offers the possibility to power this node through batteries, giving the node robustness and facility of installation, considering that agricultural fields are not characterized by having a wide electrical coverage. Although the batteries ensure the operation of the nodes, these have a limited autonomy which leads to the need for its replacement on a regular basis, increasing the need for maintenance of the system. To solve this situation, and with the aim of reducing the need for maintenance of the system, it was studied the installation of solar panels with the capacity to recharge the batteries that supply the nodes and ensure their operation throughout the year, taking into account that solar exposure differs during the year.

For this study a consumption analysis was made, for the Sensor Node in different situations. The consumption values of each component can be observed in Table 5.1.

TABLE 5.1. Components Consumption

| **Sensors** | **Number** | **Operating Current [mA]** | **Total [mA]** |
|---|---|---|---|
| Soil Moisture Sensor | 2 | 5 | 10 |
| Si7021 | 1 | 0.18 | 0.18 |
| DS18B20 | 2 | 1 | 2 |
| ESP32+RFM95W transceiver | 1 | 80 | 80 |
| **Total** | | | 92.18 |

Through the analysis of the presented table is possible to conclude that the electrical current of the node in his two states, being 0.1 mA (Table 5.2) in Deep Sleep mode and 92.18 mA in active mode.

In order to proceed to the calculations, it was established the frequency of data collection, every 10 minutes (600 seconds), and the time that the node takes to collect and send the data, 4 seconds.

Other values to consider, to proceed to the calculations, was the batteries used, which for this system was 3.7V LiPo battery with 4000 mAh (14400000 mAs).

Regarding to the calculation to find the correct size of the solar panel to use, firstly it was searched the annual average solar irradiation of the city where the node was installed, in this case, Lisbon [**51**]. The value for this city is 1713 kWh/m$^2$. In order to understand the solar irradiance for square meter, this value was divided by the number of hours of a year, shown in Equation 5.1, being this value the one that is collected by the solar panel.

$$\text{SolarRadiation} = \frac{1713\text{kWh/m}^2}{8760\text{hours}} = 196\text{W/m}^2 \tag{5.1}$$

However, the solar panels efficiency is known as low, achieving only about 15% of the value calculated before and, this same efficiency is also affected by the charging module and the charging process itself, with only about 10% of the solar irradiance being used at the end of the process [**52**], in this case, 19.6 W/m$^2$.

As explained, the sensor node worked in two different modes, having each one a specific consumption, so it was calculated the power needed to give to the node in order to work correctly. The power can be calculated using $P = U \times I$, where $P$ is the electrical power in Watts, $U$ is the electrical potential in Volts and $I$ the electrical current in Ampers. Using the power value is possible to calculate the needed size of solar panel for each situation through $S = \frac{P}{A}$, where $S$ is the size of the panel in $m^2$, $P$ is the electrical power in Watts and $A$ the average electrical power by area unit in Watts per square meters ($W/m^2$). Table 5.2 shows the values for power and panel size, for each of the node mode.

Table 5.2. Power and Solar Panel Size

| **Mode** | **U [V]** | **I [A]** | **P [W]** | **S [cm$^2$]** |
|---|---|---|---|---|
| Deep Sleep | 3.3 | 1x10$^{-4}$ | 3.3x10$^{-4}$ | 0.153 |
| Active | 3.3 | 9.218x10$^{-2}$ | 0.30419 | 155 |

The results given by the presented equations shown that the solar panel area for Deep Sleep mode is minimal, 0.153 cm$^2$, and the area for active mode is 155 cm$^2$.

Considering that every cycle last 604 seconds and only 4 seconds is related to active mode, about 0.7%, and with the aim of creating some margin, it was considered 20% of the calculated area for active mode, 31 cm$^2$.

The efficiency of a solar panel is mostly influenced by the daily solar exposure, being this variable through the year, since in the winter days, the solar exposure decreases in a large scale. To solve this situation and ensure the power supply of the node among the entire year, it was studied the duration of the used batteries. For this study it was calculated the systems consumption in both modes, Active and Deep Sleep, given the sensors and shield consumption and the duration of each mode, so it was done the presented calculations (Equations 5.2 and 5.3):

$$\mathrm{C_{DeepSleep}} = 0.1\mathrm{mA} \times 600\mathrm{s} = 60\mathrm{mAs} \tag{5.2}$$

$$\mathrm{C_{CActive}} = 92.18\mathrm{mA} \times 4\mathrm{s} = 368\mathrm{mAs} \tag{5.3}$$

Given the system consumption in both modes and the battery capacity, was then possible to calculate the number of cycles that the battery supports, through Equation 5.4.

$$\mathrm{N_{Cycles}} = \frac{\mathrm{Cap_{battery}}}{\mathrm{C_{DeepSleep}} + \mathrm{C_{Active}}} = 33645 \tag{5.4}$$

Considering that each cycle last 604 seconds, it was possible to calculate the number of cycles per hours, which is 5.96 cycles.

Finally, it was possible, through Equation 5.5, to calculate the duration of the chosen battery in days, given the number of cycles per hour and the number of cycles that the chosen battery supports.

$$\mathrm{N_{Days}} = \frac{\mathrm{N_{Cycles}}}{5.96 \times 24\mathrm{hours}} = 235\mathrm{days} \tag{5.5}$$

It was then possible to understand that through the use of a solar panel with about 31 cm$^2$ of area and a battery of 4000 mAh, it was possible to ensure the correct power supply of the developed sensor nodes, ensuring that the supply system could give the needed power to supply the node over 235 days of no solar exposure.

## 5.2. Real Case Tests

In order to understand how the system performs when exposed to real environment, the system prototype was installed in a home garden.

### 5.2.1. Implementation

The garden management was made by the owner, who watered the garden once a day, around 20:00, using a hose, being the only days that the garden was not irrigated, the rainy ones. In Figure 5.8 can be seen the garden used for the test.

FIGURE 5.8. Tested Garden

For the implementation of the final system prototype it was followed the same data collection strategy followed by the test made on the collection in real environment, in Section 5.1.2. So, the managed field was divided in two irrigation zones having each one being assigned the following sensors: air temperature and humidity, and soil moisture and temperature sensors. Besides that, the data of wind speed, wind direction and the rainfall were also collected by the weather station. The sensors implementation on the field can be seen in Figure 5.9.

Regarding to irrigation, it was installed one Actuator Node that managed two water pumps, each one attached to each irrigation zone, as seen in Figure 5.10, including also the weather station, as already demonstrated in Figure 5.3.

Figure 5.9. System Implementation - Sensor Nodes

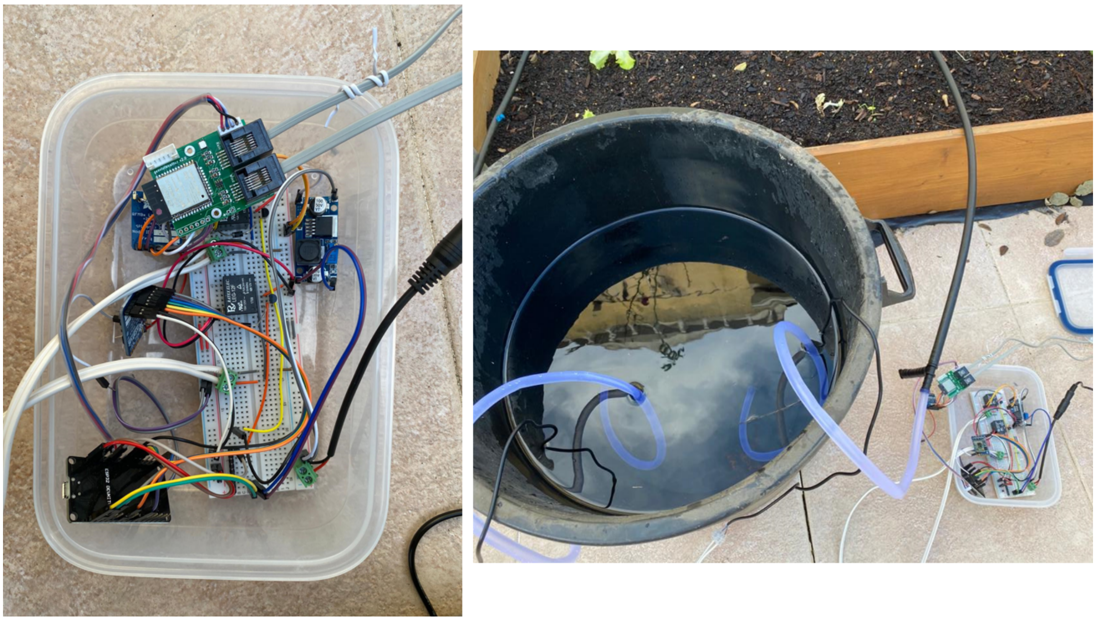

Figure 5.10. System Implementation - Actuator Node

In order to inquire the developed methods, in zone one was implemented the Watering Algorithm (*WateringAlgorithm.py*), which verifies every hour if there are some zone to be irrigated in that hour, in case of matching, the script calculates the amount of water needed for that zone. Since the owner usually irrigates his garden once a day, with the exception of rainy days, at the same hour, it was programmed that the irrigation zone that this algorithm was taking care of would be irrigated at 20:00, every day, according to the algorithm decisions.

In zone two, it was implemented the Watering Algorithm (*WateringAlgorithm.py*) and the developed Machine Learning algorithm (*ML.py*). By implementing these two algorithms in parallel, the field was autonomous, since neither the irrigation hour nor the irrigation time had to be entered manually into the system.

In order to expose the system to different situations, the prototype was in operation for a month and a half where it was exposed to good weather and rainy days.

The main goal of the presented test was to estimate the amount of water that can be saved through the implementation of the developed solutions, always ensuring the quality of the planting conditions, and also compare to the usual amount of water used manually by the user.

In order to analyse the water usually used by the owner, it was used a water flow sensor that was installed in the hose used to irrigate the garden. Through the collected values, it was possible to calculate the amount of water usually used by the owner to irrigate his garden.

### 5.2.2. Results & Discussion

The carried out tests followed several steps, in a first moment the amount of water spent by the user in each irrigation was analysed and it was concluded that the hose has a water flow of 23 liters of water per minute and, to water each irrigation zone, the user takes 25 seconds to do it. So, through the calculation $23l/min \times 25s = 5.75l$, is possible to understand that it is used 5.75 liters of water to irrigate each irrigation zone under test per day.

In a second moment an analysis was made on three distinct situations to which the system was exposed, such as: sunny days, rainy days and days after rainy days. This allows to understand how the system and each approach react to different situations. For each analysis 3 days were taken into account. For each of these days Table 5.3 presents the time irrigated and the water used. The presented results were obtained based on the water flow of the used water pumps, which was of 5.83 l/min.

Firstly it was analysed the system behaviour for the sunny days. In Figure 5.11 is possible to observe the Soil Moisture values for both zones and the moments that irrigation took place by indication of each developed algorithms.

In order to make easier the process of analysing results, the tested method in irrigation zone number one (Watering Algorithm) was identified by number 1 and the method tested in irrigation zone number two (Watering Algorithm + Machine Learning Algorithm) was identified by number 2.

TABLE 5.3. Irrigation Times

| | Method 1 | | Method 2 | |
|---|---|---|---|---|
| Day | Irrigation Time(s) | Water Usage(l) | Irrigation Time(s) | Water Usage(l) |
| 04/10 | 32 | 3.11 | - | - |
| 05/10 | 24 | 2.33 | 52 | 5.05 |
| 06/10 | 24 | 2.33 | - | - |
| 19/10 | - | - | - | - |
| 20/10 | - | - | - | - |
| 21/10 | - | - | - | - |
| 23/10 | 19 | 1.85 | - | - |
| 24/10 | 16 | 1.55 | - | - |
| 25/10 | 18 | 1.75 | 49 | 4.76 |

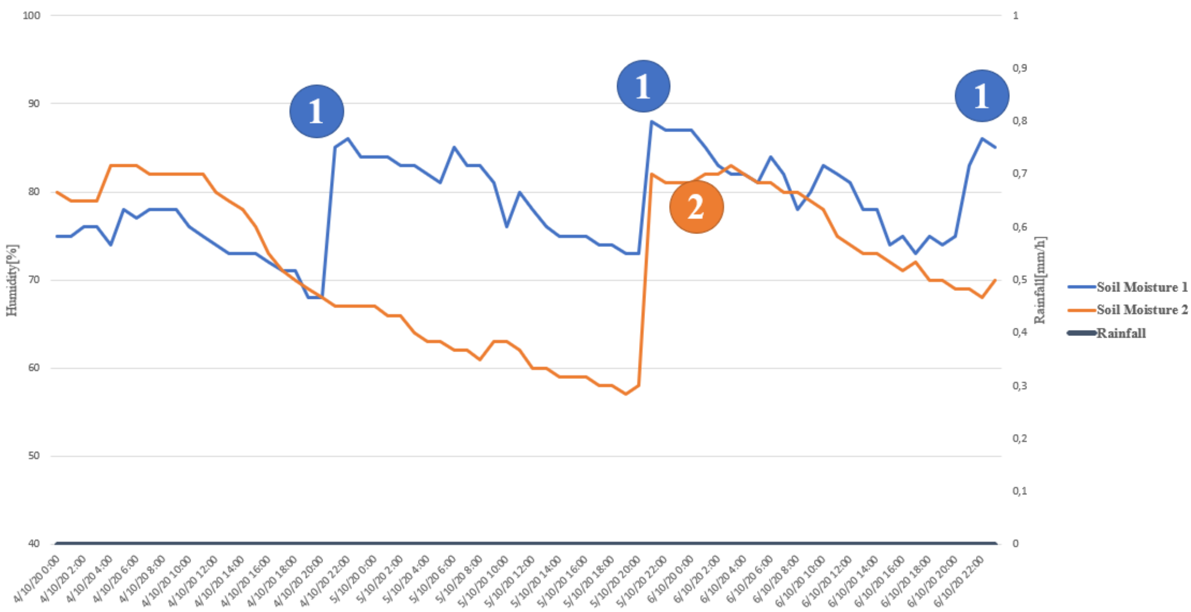


FIGURE 5.11. Sunny Days Results Graph

Through the analysis of the graph presented in Figure 5.11 and Table 5.3 it is possible to notice that during the three analysed days, the Machine Learning algorithm only pointed out the need of water administration for one of the days, also indicating that the best hour of the day to administrate that water was 20:00, having been no need of water administration for the other two days, using 52 seconds of irrigation and 5.05 l of water. The Watering Algorithm irrigated every day with a combined time of 80 seconds (32, 24, 24) and 7.77 l of water used. This shows that on a sunny day the system using the machine learning approach saves 53% more water that when using the watering and 44% more than irrigating manually.

In a second moment, the behaviour of the system for rainy days was analysed. In order to understand what was the system performance, it was created the graph presented in Figure 5.12 where can be observed the soil moisture values variations of the two irrigation zones under test and the rainfall values.

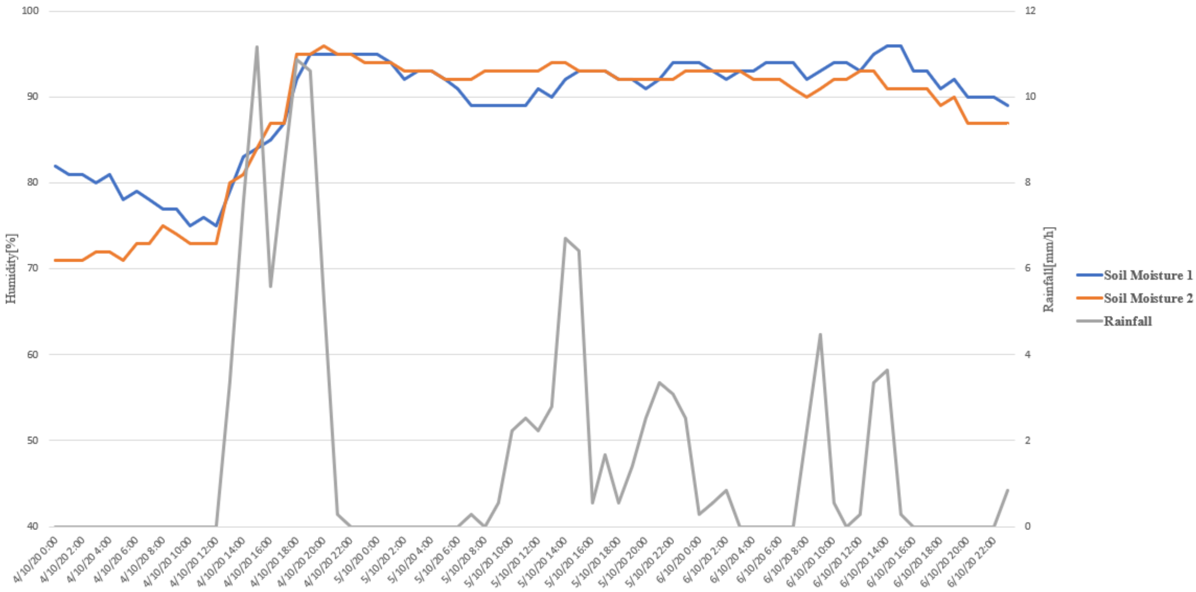


FIGURE 5.12. Rainy Days Results Graph

Having the Figure 5.12 as a basis, it is possible to notice that neither the Machine Learning nor the watering algorithms indicate the need for water administration for any of the presented hours of the rainy days, since, as can be seen in the graph, these days had long period of rain which led to an increase in soil moisture values.

In a third and last moment, the behaviour of the two implemented methods for days after rainy days was analysed. The results obtained can be seen in Figure 5.13.

Through the analysis of the presented graph, it is possible to notice that the Machine Learning algorithm only indicated the need for water administration on the third day without rain, also indicating that the best time for it to be administered was at 20:00, with 49 seconds of irrigation and 4.76 l of water used. The watering algorithm irrigated each day for a combined time of 53 seconds and 5.15 l of water. This shows that the machine learning algorithm, in a day after raining, saves 8% more water than the watering algorithm and 53% more that the manual irrigation.

As can be observed, the irrigation times for each situation depend on what the system was exposed, allowing to understand that the correct use of water vary from day to day and is important to make those changes each day to create a more sustainable irrigation process. Table 5.4 presents the results for the entire test, the month and a half, based on

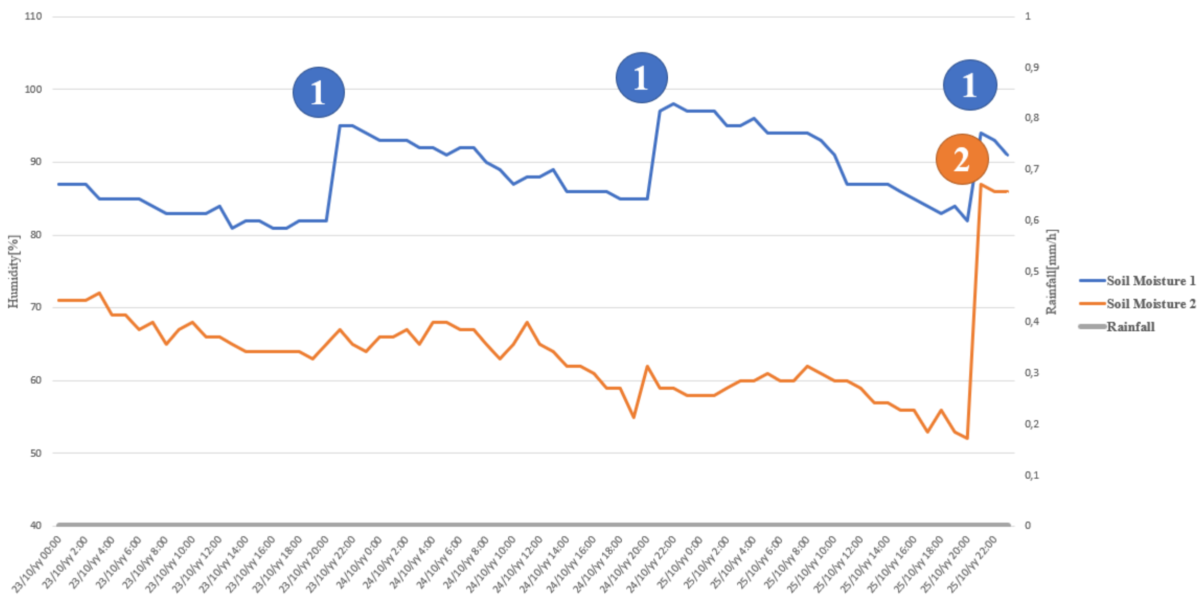


FIGURE 5.13. Days After Rainy Days Graph

water used to irrigate the field in each exposed situation, where the third method is the one used by the owner to irrigate his garden.

TABLE 5.4. Water Usage

| **Method** | **Number Days** | **Avg. Water used per day [l]** | | | **Total Consumption [l]** | | |
|---|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **1** | **2** | **3** |
| **Sunny** | 32 | 3.03 | 2.22 | 5.75 | 96.96 | 71.04 | 184 |
| **Rainy** | 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| **After Rain** | 5 | 2.05 | 2.1 | 5.75 | 10.25 | 10.5 | 28.75 |
| **Total** | 42 | | | | 107.21 | 81.54 | 212.75 |

Through the analysis of Table 5.4 is possible to understand that the method which had a minor water consumption was the one that implemented both developed algorithms, using machine learning and the watering solution, understanding the water need for the three tested situations and adjusting the amount of water administrated for each scenario.

Regarding to the first method, is possible to understand that without the hour indication for watering, this method does not achieve the best possible results since it cannot understand which were the weather conditions, needing a hour indication to perform irrigation every day. However, excluding the rainy days, the water values calculated by the watering solution, were lower than the used ones by the owner with his daily manual irrigation, by 49.6%, saving more than 26 l of water in 42 days.

It is then possible to conclude that, through the use of the combination of the two developed algorithms, was possible to save around 61.6% of water when compared to water used by the owner initially, saving more than 130 l of water in 42 days, and 23.9% when compared to the other method.

Besides the achieved water saving values, it was also possible to ensure that the system in any moment allowed the soil moisture values to reach the minimum recommended values, being this 40%, taking into account that the area where the system was tested is characterized by having a high humidity [**53**].

CHAPTER 6

# Conclusions and Future Work

## 6.1. Main Conclusions

In this dissertation it was presented an IoT solution for managing agriculture fields using a WSN and Machine Learning algorithms in order to achieve the best watering strategy. During the solution development, the project followed many steps since the analysis of the best communication solutions, the best IoT modules and the study of machine learning algorithms to understand which one best suit for the intended purpose and the final prototype test. In the end of all the presented phases it was possible to achieve a simple, versatile, user-friendly, sustainable and efficient solution.

Regarding to the communication solution used, which regards to the communication between the field installed nodes, it was possible to ensure the correct communication, ensuring that there were no messages lost through the ACK confirmation messages. With regards to the communication between the Aggregation Node to the server, a recent technology was used, the NB-IoT, and it ensured a stable communication between each part and no loss of signal or messages has been verified during the entire study.

Since that the sustainable concern was one of the principal bases of the developed solution, the implementation of solar panels in the Sensor Nodes in order to charge the used batteries through only green energy was one of the highlights of the study where it was possible to ensure a huge autonomy of each Sensor Node since that there were no need to replace any energy supply regularly like batteries.

As far as the Machine Learning is concern, one of the developed features for the system was the self-management at the irrigation level, ensuring the prediction of the best hour of the day for irrigation. For that, several classification models were tested and compared, allowing to conclude that, not only XGBoost has the accuracy for this type of task, but also that is important to test and fit the models based on the dataset and output needed. Besides that, it was also developed, through mathematical calculations, a script that returns the appropriate amount of water to be administered to each irrigation zone. Through the implementation of both scripts it was possible to achieve the pretended solution with regards to the self-manage and was also achieve a sustainable solution since the system shown the capacity to achieve of saving water values of 60%, when comparing to a manual irrigation and 23% when compared with the work of [**47**], with only a mathematical approach.

It is then possible to conclude that the developed system achieved the proposed goals, since it was possible to reach sustainability through an IoT solution using WSN and Machine Learning technologies, presenting a low maintenance requirement system with a low complexity level and high operating capacity when exposed to various weather conditions, and even give control to the user over an mobile application, having all the necessary features and requirements to be adapted in order to be introduced in the market.

## 6.2. Future Work

In the presented dissertation, the goal of the proposed system was achieved, however there are always points that can be improved and innovations to be developed as future work, such as:

- **Application Alerts:** in agricultural fields there are some risk situation that, when not quickly resolved, can lead to crop compromising. In this way, risk situation alerts were a point to take into count for the improvement of the system and the mobile application;
- **Water Pumps:** the used water pumps were dimensioned for a home garden, being these not suitable for larger fields, thus becoming the interesting factor to be improved;
- **Web Site:** for the implementation of the developed system in larger fields where there are several people responsible for its maintenance, it would be interesting to develop a web site to manage both the irrigation zones and the actuators as well as the users;
- **Public Gardens:** since the environmental concern is increasing, and the municipalities are concerned about it, it would be interesting to adapt the system developed for public gardens;
- **Large scale application:** the presented system was implemented in a small scale, only using one node of each created type. In order to achieve even more resource savings, a challenge for this system would be to be installed in large agricultural fields where there is a need for many sensor and actuator nodes.

# References


[1] Food and Agriculture Organization of the United Nations, "Water for Sustainable Food and Agriculture," [Online] Available: http://www.fao.org/3/a-i7959e.pdf, (visited 10/03/2020).

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] L. Shkurti, X. Bajrami, E. Canhasi, B. Limani, S. Krrabaj, and A. Hulaj, "Development of ambient environmental monitoring system through wireless sensor network (WSN) using NodeMCU and 'WSN monitoring'," *2017 6th Mediterranean Conference on Embedded Computing, MECO 2017 - Including ECYPS 2017, Proceedings*, no. June, pp. 1–5, 2017.

[4] J. E. Siegel, S. Kumar, and S. E. Sarma, "The future internet of things: Secure, efficient, and model-based," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2386–2398, 2018.

[5] B. Ahlgren, M. Hidell, and E. C. Ngai, "Internet of Things for Smart Cities: Interoperability and Open Data," *IEEE Internet Computing*, vol. 20, no. 6, pp. 52–56, 2016.

[6] M. Burhan, R. A. Rehman, B. Khan, and B. S. Kim, "IoT elements, layered architectures and security issues: A comprehensive survey," *Sensors (Switzerland)*, vol. 18, no. 9, 2018.

[7] M. Guo, J. Nowakowska-Grunt, V. Gorbanyov, and M. Egorova, "Green technology and sustainable development: Assessment and green growth frameworks," *Sustainability (Switzerland)*, vol. 12, no. 16, 2020.

[8] J. Ruan, Y. Wang, F. T. S. Chan, X. Hu, M. Zhao, F. Zhu, B. Shi, Y. Shi, and F. Lin, "A Life Cycle Framework of Green IoT-Based Agriculture and Its Finance, Operation, and Management Issues," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 90–96, 2019.

[9] A. Mondal, I. S. Misra, and S. Bose, "Building a low cost solution using wireless sensor network for agriculture application," *Proceedings of 2017 International Conference on Innovations in Electronics, Signal Processing and Communication, IESC 2017*, pp. 61–65, 2017.

[10] C. Kamienski, J. P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T. S. Cinotti, R. F. Maia, and A. T. Neto, "Smart water management platform: IoT-based precision irrigation for agriculture," *Sensors (Switzerland)*, vol. 19, no. 2, 2019.

[11] M. K. Gayatri, J. Jayasakthi, and G. S. Mala, "Providing Smart Agricultural solutions to farmers for better yielding using IoT," *Proceedings - 2015 IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development, TIAR 2015*, no. Tiar, pp. 40–43, 2015.

[12] J. Treboux and D. Genoud, "High Precision Agriculture: An Application of Improved Machine-Learning Algorithms," *Proceedings - 6th Swiss Conference on Data Science, SDS 2019*, no. Ml, pp. 103–108, 2019.

[13] C. Dionisio, P. Sebastiao, and N. Souto, “Distributed sensing solution for home efficiency tracking,” Master’s thesis, ISCTE-Instituto Universitário de Lisboa, 2019.

[14] G. Simoes, P. Sebastiao, and N. Souto, “Smart System for Monitoring and Control of Swimming Pools,” Master’s thesis, ISCTE-Instituto Universitário de Lisboa, 2019.

[15] D. Elmazi, M. Cuka, T. Oda, M. Ikeda, and L. Barolli, “Effect of node density on actor selection in wsans: A comparison study for two fuzzy-based systems,” *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, pp. 865–871, 2017.

[16] Espressif, “ESP32 Series Datasheet,” pp. 1–61, [Online] Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, (visited 03/01/2020).

[17] A. A. Galadima, “Arduino as a learning tool,” *Proceedings of the 11th International Conference on Electronics, Computer and Computation, ICECCO 2014*, pp. 14–17, 2014.

[18] R. Pi, “RaspberryPi 3 Datasheet,” [Online] Available: https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf, (visited 20/03/2020).

[19] A. Maier, A. Sharp, and Y. Vagapov, “Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things,” *2017 Internet Technologies and Applications, ITA 2017 - Proceedings of the 7th International Conference*, pp. 143–148, 2017.

[20] A. Gloria, F. Cercas, and N. Souto, “Comparison of communication protocols for low cost Internet of Things devices,” *South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2017*, 2017.

[21] J. Cai and O. C. Ugweje, “A comparative study of wireless ATM MAC protocols,” *2002 International Conference on Communications, Circuits and Systems and West Sino Exposition, ICCCAS 2002 - Proceedings*, pp. 516–520, 2002.

[22] D. Hatvani and D. Macko, “Increasing Bluetooth Low Energy communication efficiency by presetting protocol parameters,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 5, pp. 3486–3499, 2019.

[23] L. Feltrin, M. Condoluci, T. Mahmoodi, M. Dohler, and R. Verdone, “NB-IoT: Performance estimation and optimal configuration,” *24th European Wireless 2018 ”Wireless Futures in the Era of Network Programmability”, EW 2018*, pp. 50–55, 2018.

[24] S. S. I. Samuel, “A review of connectivity challenges in IoT-smart home,” *2016 3rd MEC International Conference on Big Data and Smart City, ICBDSC 2016*, pp. 364–367, 2016.

[25] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of Lora: Long range & low power networks for the internet of things,” *Sensors (Switzerland)*, vol. 16, no. 9, 2016.

[26] Bharati Wukkadada ; Kirti Wankhede ; Ramith Nambiar ; Amala Nair, “Comparison with HTTP and MQTT In Internet of Things (IoT) - IEEE Conference Publication,” *2018 International Conference on Inventive Research in Computing Applications (ICIRCA 2018)*, no. Icirca, pp. 249–253, 2018. [Online]. Available: https://ieeexplore-ieee-org.e.bibl.liu.se/document/8597401

[27] D. Ghosh, A. Agrawal, N. Prakash, and P. Goyal, "Smart saline level monitoring system using ESP32 and MQTT-S," *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services, Healthcom 2018*, pp. 1–5, 2018.

[28] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019. [Online]. Available: https://doi.org/10.1016/j.icte.2017.12.005

[29] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: Approaches and challenges," *arXiv*, no. February, pp. 94–101, 2018.

[30] D. Praveen Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Information Fusion*, vol. 49, no. September 2018, pp. 1–25, 2019. [Online]. Available: https://doi.org/10.1016/j.inffus.2018.09.013

[31] A. Dave, "Regression in Machine Learning," [Online] Available: https://medium.com/datadriveninvestor/regression-in-machine-learning-296caae933ec, (visited 10/10/2020).

[32] A. Goldstein, L. Fink, A. Meitin, S. Bohadana, O. Lutenberg, and G. Ravid, "Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge," *Precision Agriculture*, vol. 19, no. 3, pp. 421–444, 2018.

[33] S. Asiri, "Machine Learning Classifiers," [Online] Available: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623, (visited 10/10/2020).

[34] Scikit-learn, "Random Forest Classifier," [Online] Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, (visited 27/08/2020).

[35] ——, "Decision Tree Classifier," [Online] Available: https://scikit-learn.org/stable/modules/tree.html, (visited 27/08/2020).

[36] F. J. Yang, "An extended idea about decision trees," *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*, pp. 349–354, 2019.

[37] Scikit-learn, "Support Vector Machines," [Online] Available: https://scikit-learn.org/stable/modules/svm.html, (visited 27/08/2020).

[38] ——, "Neural Networks Supervised," [Online] Available: https://scikit-learn.org/stable/modules/neural{_}networks{_}supervised.html, (visited 27/08/2020).

[39] XGBoost, "XGBoost," [Online] Available: https://xgboost.readthedocs.io/en/latest/parameter.html, (visited 27/08/2020).

[40] J. Li and R. Zhang, "Dynamic Weighting Multi Factor Stock Selection Strategy Based on XGboost Machine Learning Algorithm," *Proceedings of 2018 IEEE International Conference of Safety Produce Informatization, IICSPI 2018*, pp. 868–872, 2019.

[41] Silicon Labs, "Si7021-A20 Dataset," [Online] Available: https://www.silabs.com/documents/public/data-sheets/Si7021-A20.pdf, (visited 19/09/2020).

[42] Dallas, "DS18B20," [Online] Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/227472/DALLAS/DS18B20.html, (visited 19/09/2020).

[43] DFRobot, "Capacity Soil Moisture Sensor," [Online] Available: https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193, (visited 10/09/2020).

[44] ——, "Wheather Station," [Online] Available: https://wiki.dfrobot.com/Weather_Station_with_Anemometer_Wind_vane_Rain_bucket_SKU_SEN0186, (visited 06/09/2020).

[45] A. Fuad, A. Erwin, and H. P. Ipung, "Processing performance on Apache Pig, Apache Hive and MySQL cluster," *Proceedings of 2014 International Conference on Information, Communication Technology and System, ICTS 2014*, pp. 297–301, 2014.

[46] IPMA, "IPMA - API," [Online] Available: http://api.ipma.pt/, (visited 15/08/2020).

[47] A. Glória, P. Sebastião, C. Dionísio, G. Simões, and J. Cardoso, "Water management for sustainable irrigation systems using internet-of-things," *Sensors (Switzerland)*, vol. 20, no. 5, pp. 1–14, 2020.

[48] Scikit-learn, "RandomizedSearchCV," [Online] Available: https://scikit-learn.org/stable/modules/generated/sklearn.model{_}selection.RandomizedSearchCV.html, (visited 27/08/2020).

[49] ——, "scikit-learn," [Online] Available: https://scikit-learn.org/stable/, (visited 27/08/2020).

[50] S. Lu, B. Wang, H. Wang, and Q. Hong, "A hybrid collaborative filtering algorithm based on KNN and gradient boosting," *13th International Conference on Computer Science and Education, ICCSE 2018*, no. Iccse, pp. 432–436, 2018.

[51] Solargis, "Solargis," [Online] Available: https://solargis.info, (visited 30/10/2020).

[52] T. G. Patil and S. Asokan, "Comparative analysis of calculation of solar panel efficiency degradation," *ICONSTEM 2017 - Proceedings: 3rd IEEE International Conference on Science Technology, Engineering and Management*, vol. 2018-January, pp. 522–527, 2017.

[53] Food and Agriculture Organization of the United Nations, "Optimizing soil moisture for plant production," [Online] Available: http://www.fao.org/3/a-y4690e.pdf, (visited 20/09/2020).

[54] ——, "Single crop coefficient (Kc)," [Online] Available: http://www.fao.org/3/x0490e/x0490e0b.htm, (visited 18/10/2020).

[55] ——, "Meteorological tables," [Online] Available: http://www.fao.org/3/X0490E/x0490e0j.htm, (visited 18/10/2020).

# Appendices

APPENDIX A

# Crop Coeficient, Incident Radiation and Configuration Parameters Tables

In Table A1 is presented the Crop Coefficient, $K_c$, values of the most cultivated foods in Mediterranean zone [**54**]. The table presents three values, as crops might need different $K_c$ values as they grow.

TABLE A1. Crop Coefficient $K_c$ Values

| **Crop** | **Kc(ini)** | **Kc (mid)** | **Kc(end)** |
|---|---|---|---|
| **Broccoli** | | 1.05 | 0.95 |
| **Brussel Sprouts** | | 1.05 | 0.95 |
| **Cabbage** | | 1.05 | 0.95 |
| **Carrots** | | 1.05 | 0.95 |
| **Cauliflower** | | 1.05 | 0.95 |
| **Celery** | | 1.06 | 1.00 |
| **Garlic** | | 1.00 | 0.70 |
| **Lettuce** | | 1.00 | 0.95 |
| **Onions** | | 1.00 | 1.00 |
| **Spinach** | | 1.00 | 0.95 |
| **Radish** | | 0.90 | 0.85 |
| **Tomato** | | 1.15 | 0.70-0.90 |
| **Cucumber** | | 1.00 | 0.75 |
| **Watermelon** | | 1.00 | 0.75 |
| **Potato** | | 1.15 | 0.75 |
| **Beans, green** | 0.5 | 1.05 | 0.90 |
| **Beans, dry and Pulses** | 0.4 | 1.15 | 0.35 |
| **Rice** | 1.05 | 1.20 | 0.90-0.60 |

Table A2 presents the values for the incident extra-terrestrial solar radiation for the Northern Hemisphere, $R_0$, which are based on latitude and time of the year [**55**]. The values are presented in $MJm^{-2}day^{-1}$ and Equation A.1 can be used to convert them into $mm/day$.

$$R_0[mm/day] = R_0[MJm^{-2}day^{-1}] \times \frac{238.85}{(597.3 - 0.57 \times temp_{avg})} \tag{A.1}$$

Table A2. Incident Extra-terrestrial Solar Radiation Values

| Lat. Deg. | Jan | Feb | Mar | Apr | May | Jun | July | Aug | Sep | Oct | Nov | Dec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **70** | 0.0 | 2.6 | 10.4 | 23.0 | 35.2 | 42.5 | 39.4 | 28.0 | 14.9 | 4.9 | 0.1 | 0.0 |
| **68** | 0.1 | 3.7 | 11.7 | 23.9 | 35.3 | 42.0 | 38.9 | 28.6 | 16.1 | 6.0 | 0.7 | 0.0 |
| **66** | 0.6 | 4.8 | 12.9 | 24.8 | 35.6 | 41.4 | 38.8 | 29.3 | 17.3 | 7.2 | 1.0 | 0.1 |
| **64** | 1.4 | 5.9 | 14.1 | 25.8 | 35.9 | 41.2 | 38.8 | 20.0 | 18.4 | 8.5 | 2.4 | 0.6 |
| **62** | 2.3 | 7.1 | 15.4 | 26.6 | 36.3 | 41.2 | 39.0 | 30.6 | 19.5 | 9.7 | 3.4 | 1.3 |
| **60** | 3.3 | 0.3 | 6.5 | 27.5 | 36.6 | 41.2 | 39.2 | 31.3 | 20.6 | 10.9 | 4.4 | 2.2 |
| **58** | 4.3 | 9.6 | 17.7 | 28.4 | 37.0 | 41.3 | 39.4 | 32.0 | 21.7 | 12.1 | 5.5 | 3.1 |
| **56** | 5.4 | 10.8 | 18.9 | 29.2 | 37.4 | 41.4 | 39.6 | 32.6 | 22.7 | 13.3 | 6.7 | 4.2 |
| **54** | 6.5 | 12.0 | 20.0 | 30.0 | 37.8 | 41.5 | 39.8 | 33.2 | 23.7 | 14.5 | 7.8 | 5.2 |
| **52** | 7.7 | 13.2 | 21.1 | 30.8 | 38.2 | 41.6 | 40.1 | 33.8 | 24.7 | 15.7 | 9.0 | 6.4 |
| **50** | 8.9 | 14.4 | 22.2 | 31.5 | 38.5 | 41.7 | 40.2 | 34.4 | 25.7 | 16.9 | 10.2 | 7.5 |
| **48** | 10.1 | 15.7 | 23.3 | 32.2 | 38.8 | 41.8 | 40.4 | 34.9 | 25.6 | 18.1 | 11.4 | 8.7 |
| **46** | 11.3 | 16.9 | 24.3 | 32.9 | 39.1 | 41.9 | 40.6 | 35.4 | 27.5 | 19.2 | 12.6 | 9.9 |
| **44** | 12.5 | 18.0 | 25.3 | 33.5 | 39.3 | 41.9 | 40.7 | 35.9 | 29.4 | 20.3 | 13.9 | 11.1 |
| **42** | 13.8 | 19.2 | 26.3 | 34.1 | 39.5 | 41.9 | 40.8 | 36.3 | 29.2 | 21.4 | 15.1 | 12.4 |
| **40** | 15.0 | 20.4 | 27.2 | 34.7 | 39.7 | 41.9 | 40.8 | 36.7 | 30.0 | 22.0 | 16.3 | 13.5 |
| **38** | 16.2 | 21.5 | 28.1 | 35.2 | 29.9 | 41.8 | 40.8 | 37.0 | 30.7 | 23.6 | 17.5 | 14.8 |
| **36** | 17.5 | 22.6 | 29.0 | 5.7 | 40.0 | 41.7 | 40.8 | 37.4 | 31.5 | 24.6 | 18.7 | 16.1 |
| **34** | 18.7 | 23.7 | 29.9 | 36.1 | 40.0 | 41.6 | 40.8 | 37.6 | 31.2 | 25.6 | 19.9 | 17.3 |
| **32** | 19.9 | 24.8 | 30.7 | 36.5 | 40.0 | 41.4 | 40.7 | 37.9 | 32.8 | 26.5 | 21.1 | 18.5 |
| **30** | 21.1 | 25.8 | 31.4 | 36.8 | 40.0 | 41.2 | 40.6 | 38.0 | 33.4 | 27.6 | 22.2 | 19.8 |
| **28** | 22.3 | 25.8 | 32.2 | 37.1 | 40.0 | 40.9 | 40.4 | 38.2 | 33.9 | 28.5 | 23.3 | 21.0 |
| **26** | 23.4 | 27.6 | 32.8 | 37.4 | 39.9 | 40.6 | 40.2 | 36.2 | 34.5 | 29.3 | 24.5 | 22.2 |
| **24** | 24.6 | 28.8 | 33.5 | 37.6 | 39.7 | 40.3 | 39.9 | 38.3 | 34.9 | 30.2 | 25.5 | 23.3 |
| **22** | 25.7 | 29.7 | 34.1 | 37.8 | 39.5 | 40.0 | 39.6 | 38.4 | 35.4 | 31.0 | 26.6 | 24.5 |
| **20** | 26.8 | 30.6 | 34.7 | 37.9 | 9.3 | 39.5 | 39.3 | 38.3 | 35.8 | 31.8 | 27.7 | 25.6 |
| **18** | 27.9 | 31.5 | 35.2 | 38.0 | 39.0 | 39.1 | 38.9 | 38.2 | 35.1 | 32.5 | 28.7 | 26.8 |
| **16** | 28.9 | 32.3 | 35.7 | 38.1 | 38.7 | 38.6 | 38.5 | 38.1 | 36.4 | 33.2 | 29.6 | 27.9 |
| **14** | 29.9 | 33.1 | 36.1 | 38.1 | 38.4 | 38.1 | 38.1 | 38.0 | 36.7 | 33.9 | 30.6 | 28.9 |
| **12** | 30.9 | 33.8 | 36.5 | 38.0 | 38.0 | 37.6 | 37.5 | 37.8 | 36.9 | 4.5 | 31.5 | 30.0 |
| **10** | 31.9 | 34.5 | 36.9 | 37.9 | 37.6 | 37.0 | 37.1 | 37.5 | 37.1 | 5.1 | 32.4 | 31.0 |
| **8** | 32.8 | 35.2 | 37.2 | 37.8 | 37.1 | 36.3 | 36.5 | 37.2 | 37.2 | 35.6 | 33.3 | 32.0 |
| **6** | 33.7 | 35.6 | 37.4 | 37.6 | 36.6 | 35.7 | 5.9 | 36.9 | 37.3 | 36.1 | 34.1 | 32.9 |
| **4** | 34.6 | 36.4 | 7.6 | 37.4 | 36.0 | 350.0 | 35.3 | 36.5 | 37.3 | 36.6 | 34.9 | 33.9 |
| **2** | 35.4 | 37.0 | 37.8 | 37.1 | 35.4 | 34.2 | 34.6 | 36.1 | 37.3 | 37.0 | 35.6 | 34.8 |
| **0** | 36.2 | 37.5 | 37.9 | 36.8 | 34.8 | 33.4 | 33.9 | 35.7 | 37.2 | 37.4 | 36.3 | 35.6 |

Table A3 presents the default parameters of each Machine Learning algorithm used, based on their documentation [**34, 35, 37, 38, 39**].

Table A3. Default Configuration Parameters

| Algorithm | Parameter Configuration |
|---|---|
| XGBoost | $eta = 0.3, gamma = 0, max\_depth = 6, min\_child\_weight = 1, max\_delta\_step = 0, subsample = 1,$ $sampling\_method = uniform, colsample\_by* = 1, lambda = 1, alpha = 0, tree\_method = auto, sketch\_eps = 0.03,$ $scale\_pos\_weight = 1, updater = grow\_colmaker, refresh\_leaf = 1, process\_type = default, grow\_policy = depthwise,$ $max\_leaves = 0, max\_bin = 256, predictor =' auto', num\_parallel_tree = 1$ |
| Random Forest | $n\_estimators = 10, criterion =' gini', max\_depth = None, min\_samples\_split = 2, min\_samples\_leaf = 1,$ $max\_features =' auto', max\_leaf\_nodes = None, bootstrap = True, oob\_score = False, n\_jobs = 1,$ $random\_state = None, verbose = 0, min\_density = None, compute\_importances = None$ |
| Neural Network | $hidden\_layer\_sizes = (100, ), activation =' relu', *, solver =' adam', alpha = 0.0001, batch\_size =' auto',$ $learning\_rate =' constant', learning\_rate\_init = 0.001, power\_t = 0.5, max\_iter = 200, shuffle = True,$ $random_state = None, tol = 0.0001, verbose = False, warm_start = False, momentum = 0.9,$ $nesterovs\_momentum = True, early\_stopping = False, validation\_fraction = 0.1, beta\_1 = 0.9, beta\_2 = 0.999,$ $epsilon = 1e - 08, n\_iter\_no\_change = 10, max\_fun = 15000$ |
| SVM (linear) | $penalty =' l2', loss =' squared_hinge', *, dual = True, tol = 0.0001, C = 1.0, multi\_class =' ovr',$ $fit\_intercept = True, intercept\_scaling = 1, class\_weight = None, verbose = 0, random\_state = None, max\_iter = 1000$ |
| Decision Tree | $criterion =' gini', splitter =' best', max\_depth = None, min\_samples\_split = 2,$ $min\_samples\_leaf = 1, min\_weight\_fraction\_leaf = 0.0, max\_features = None,$ $random\_state = None, max\_leaf\_nodes = None, min\_impurity\_decrease = 0.0, min\_impurity\_split = None,$ $class\_weight = None, presort =' deprecated', ccp\_alpha = 0.0$ |

APPENDIX B

## Images of Main Sensors & Relational Diagram of Database

As explained in Section 3.1.3, to the sensor node were attached the following sensors:

- SI7021, an air temperature and humidity sensor;

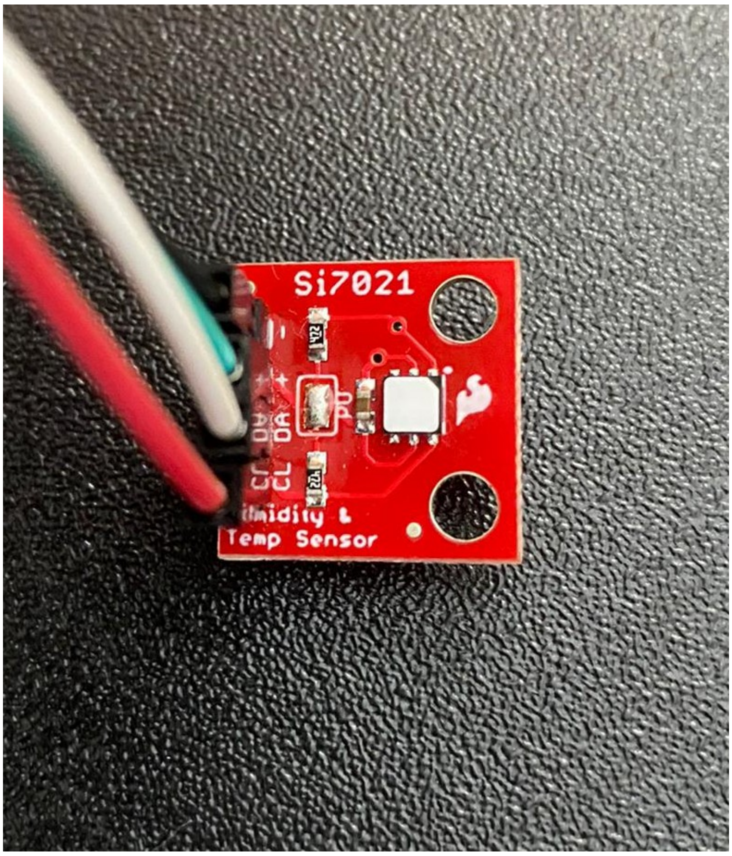


Figure B1. SI7021 Sensor

- DS18B20, a waterproof temperature sensor, used for soil temperature;
- Analog Capacity Soil Moisture, a waterproof humidity sensor for soil moisture;

Figure B4 shows the created tables and corresponding connections between keys for the developed database that supports the system.

Figure B2. DS18B20 Sensor

FIGURE B3. Analog Capacity Soil Moisture Sensor

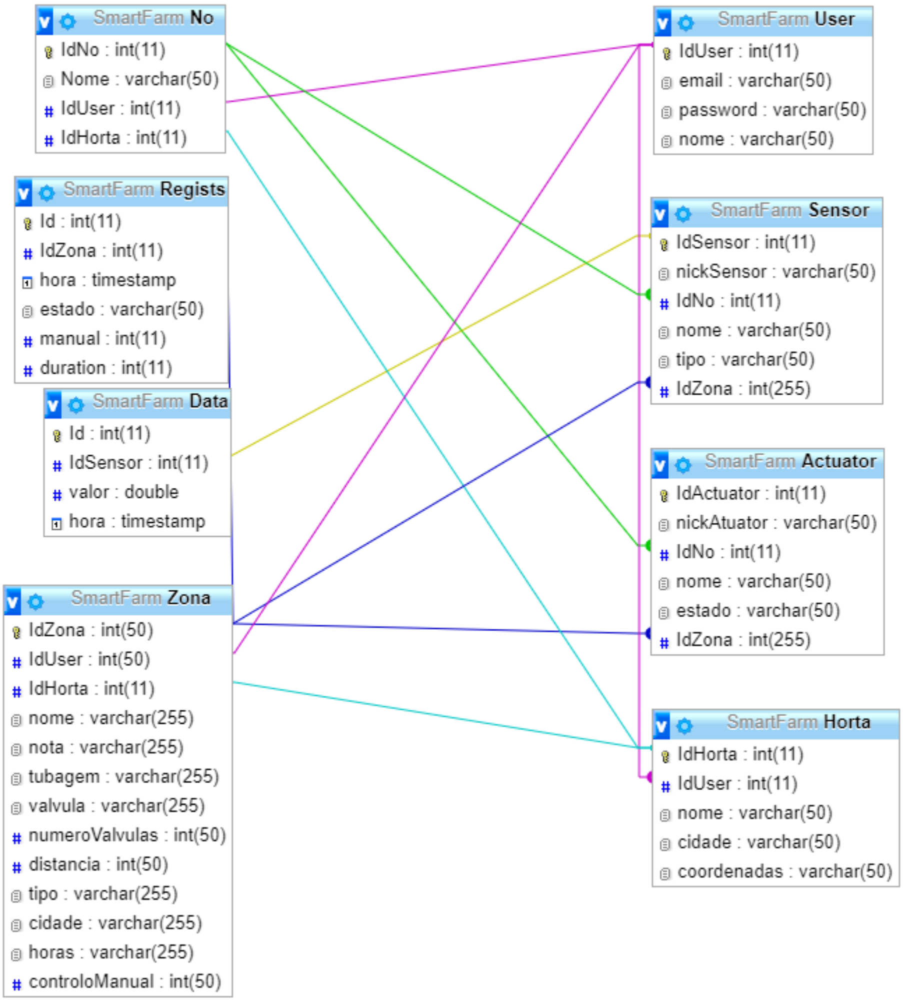


Figure B4. Database

APPENDIX C

# User & Technical Manual

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

User & Technical Manual

João Miguel Botas Cardoso

Master in Telecommunications and Computer Engineering

Supervisor
Prof. Dr. Pedro Joaquim Amaro Sebastião, Assistant Professor
ISCTE-IUL

Co-Supervisor
Master André Filipe Xavier da Glória, Researcher
ISCTE-IUL

November, 2020

## Contents

## List of Figures

CHAPTER 1

# Developed Platform

In order to give to the user the best experience possible, it was developed a mobile application for iOS so that the user had the possibility to see the collected information in real time and to perform any necessary task, just by using his mobile phone and without the need to be in site. In the presented sections are described the working process of the mobile application as well as the created solutions that were implemented on the server.

## 1.1. Mobile Application

The developed mobile application was dimensioned to serve as a management tool, having the necessary features to processed to all the necessary actions that the field owner had to has for a well manage agricultural field. For the development of this mobile application it was chosen the iOS mobile operating system. For this the application was developed in Swift, using its fifth version(Swift5), through the official IDE for Swift XCode.

In Figure 1.1 is shown the application icon which is presented among the other user's applications.


FIGURE 1.1. Application Logo

So, whenever the user needs to make a field condition enquiry or perform any necessary action, the application establish the necessary connection to the server in order to send and receive the necessary information.

In order to understand the logic of the application development, Figure 1.2 shows the application Flowchart.

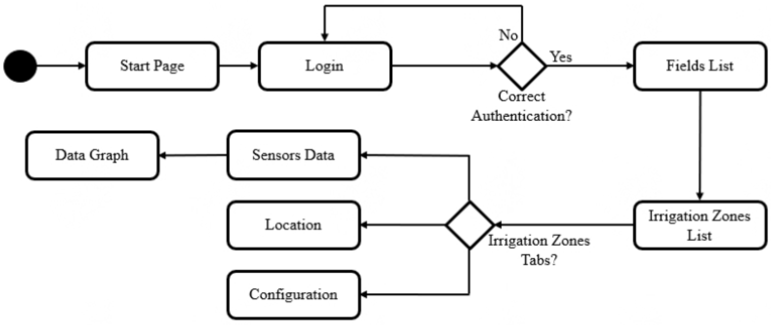


FIGURE 1.2. Application flow

### 1.1.1. Home and Authentication View

In the moment that the application is initiated is shown the Home View, presented in Figure 1.3. By clicking on the Login button the application leads the user to the authentication view, shown in Figure 1.4, where it is asked to introduce the personal credentials. In case of login error is shown an error messages as presented in Figure 1.5.


FIGURE 1.3. Authentication View

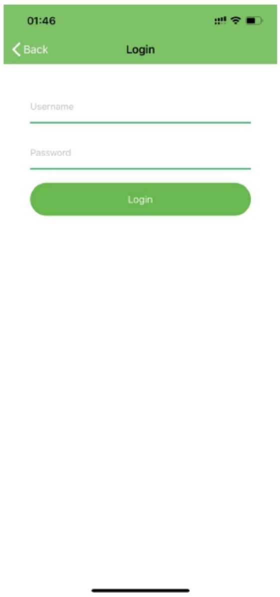


FIGURE 1.4. Home View

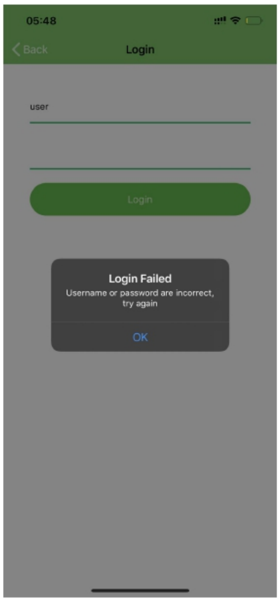


FIGURE 1.5. Login Failed Message

#### 1.1.2. List of Fields

By matching the correct login, the user is leaded to the fields view where is shown the fields that the user manage. This view is presented in Figure 1.6 where the user has to select the pretended one.

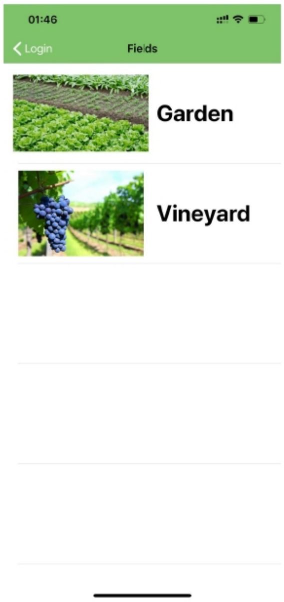


FIGURE 1.6. Fields List View

#### 1.1.3. List of Irrigation Zones

In each field there are multiple irrigation zones with specific types of crops, in order to select the pretended one is shown the Irrigation Zones List view as showed in Figure 1.7.

#### 1.1.4. Irrigation Zone Control View

By clicking in the pretended irrigation zone, the application takes the user to the Irrigation Zone Control View . In this view are presented three tabs in the bottom of the screen: Sensors, Location and Configuration.

**1.1.4.1.** ***Sensors.*** The sensors tab shows the user the last collected values from each sensor of the chosen irrigation zone (Figure 1.8). In order to offer the user the possibility of consult the graph of the data variation of each installed sensor, by clicking on the desired sensor, the application shows the user the desired graph. In this graph it is possible to

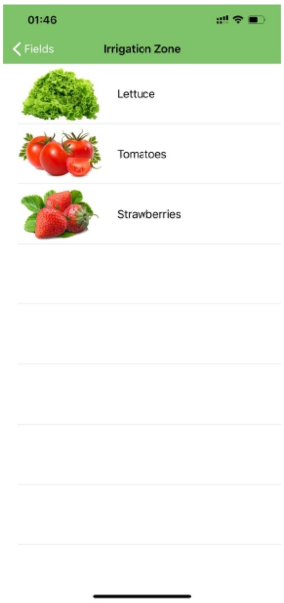


FIGURE 1.7. Irrigation Zones List View

select the data of that day, the previous one, and the maximum and minimum values of the last week (Figure 1.9).

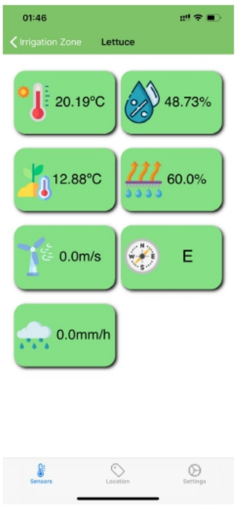


FIGURE 1.8. Values Graph

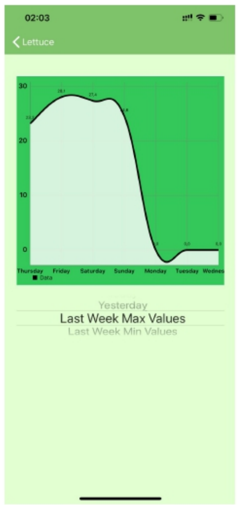


FIGURE 1.9. Values Graph

**1.1.4.2.** ***Location.*** Regarding to the location tab, this one points on the map the location of the chosen irrigation zone, as chosen in Figure 1.10.


FIGURE 1.10. Irrigation Zone Location

**1.1.4.3.** ***Configuration.*** In configuration tab (Figure 1.11) is presented to the user weather forecast, irrigation zone configuration features and the irrigation control. Regarding to the weather forecast, is presented the max and minimum temperatures values for the present day and the forecast for the next two days as well as an atmospheric conditions illustration. With regards to the irrigation zone configuration features, this ones are used to indicate all the necessary features of the developed scripts for the self management of the zone. Finally, and regarding to the irrigation control, in the bottom panel is possible to select the irrigation mode pretended, being the options Automatic or Manual, and the necessary tool to turn ON or OFF the irrigation of the selected zone, with the possibility of indicating the desired watering time. When a manual irrigation is started, is shown the time until this one will last (Figure 1.12).

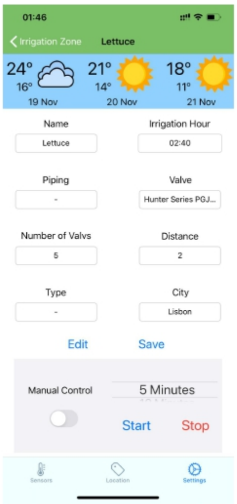


FIGURE 1.11. Configuration Tab

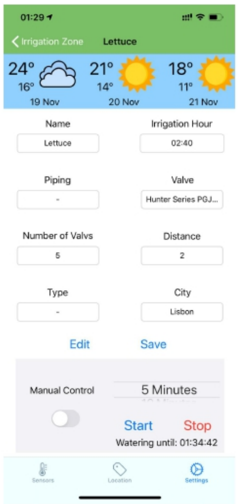


FIGURE 1.12. Watering Start

### 1.2. Server

Regarding to the backoffice, it was developed a set of solutions was developed in order to provide access to the functioning of the whole system, thus a database and algorithms in Python were created.

#### 1.2.1. Database

As far as the database is concerned, it has been divided into several tables in order to make it possible to store the information in an organized manner. In Figure 1.13 is shown the UML model of the system database.

As can be seen, the data base was divided in multiple tables, being the follow:

- **User** In this table is stored the users information's for authentication matter and for the association for each owned fields.
- **Horta** Is where is stored the fields information of each user.
- **Zona** This table is related to each irrigation zone and where is stored all the information related to it. The information stored in table is the one that can be updated through the configuration tab presented in Figure 1.11.
- **No** This table is responsible so store the information of each node.
- **Actuator** The Actuator table is the on responsible for storing the information related to each actuator. In this table the parameter *estado* is updated whenever the actuator status is changed, being 0 for *OFF* and 1 for *ON*.

- **Sensor** The Sensor table is the one responsible to store the information related to each sensor of the system.

In order to save all the changes made into the system and also all the collected data, the following tables were also considered:

- **Regists** This table is the responsible one for saving all changes made to the status of the actuators as well as the time they need to be *ON*.
- **Data** Is the table responsible to store all the collected data from each sensor. Associated to each value is the id of the sensor that collected it and the date and time that it was collected.

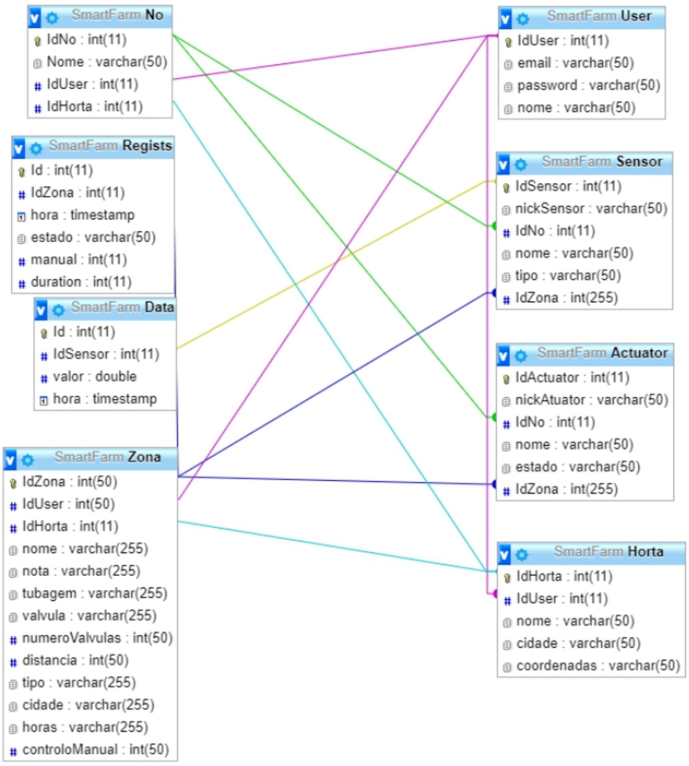


FIGURE 1.13. Database

### 1.2.2. Scrips

In order to guarantee the correct logic and functioning of the system, two types of python algorithms have been developed and run in background on the server: the ones responsible to handling the information exchange between the server and the field or the mobile application and the ones that ensure the automatic analysis of the data collected and the consequent actions to be performed.

Regarding to the developed scripts to handle the information exchange there were developed the scripts presented in Figure 1.14.

| Nome | Tamanho | Tipo | Modificado | Permissões | Proprie... |
|---|---|---|---|---|---|
| .. | | | | | |
| api.py | 0 | Ficheiro PY | 26/11/2020 03:... | -rw-rw-r-- | cardos... |
| mqttBD.py | 0 | Ficheiro PY | | | |

FIGURE 1.14. Python Handler Files

- **API** The *api.py* was the one developed to act as the system API in order to ensure the connection of the mobile application to the database through the exchange of HTTP packages.
- ***mqttBD.py*** This python script ensure the message exchange between the server and the field modules through MQTT.

Regarding to the developed algorithms responsible for field self management, those presented in Figure 1.15 have been developed:

| Nome | Tamanho | Tipo | Modificado | Permissões | Proprie... |
|---|---|---|---|---|---|
| .. | | | | | |
| ML.py | 0 | Ficheiro PY | 26/11/2020 03:... | -rw-rw-r-- | cardos... |
| WaterringAlgorithm.py | 0 | Ficheiro PY | 26/11/2020 03:... | -rw-rw-r-- | cardos... |

FIGURE 1.15. Database

- **WatteringAlgorithm.py** This script is responsible to calculate the amount of water needed to be administrated to a specific irrigation zone, based on the data entered in the irrigation zone settings, the climate forecast and the collected values.
- ***ML.py*** The *ML.py* is the developed Machine Learning algorithm responsible to predict the best hour for irrigation through the analysis of the collected data and based on all the learning that it already has, resulting from its training.

CHAPTER 2

# Prototype

This chapter is related to the developed prototype.

The prototype is composed by tree types of nodes, being necessary the implementation of one of each type so that the system can offer all the functionalities it has.

## 2.1. Aggregation Node

Regarding to the Aggregation Node, this one is the center of the field implementation system, being the point which all information flows. This one need to be installed in the side so that can establish communications with all the installed network and then communicate with the server through mobile network. In Figure 2.2 is shown the node with his modules and in Figure 2.1 is presented its circuit.

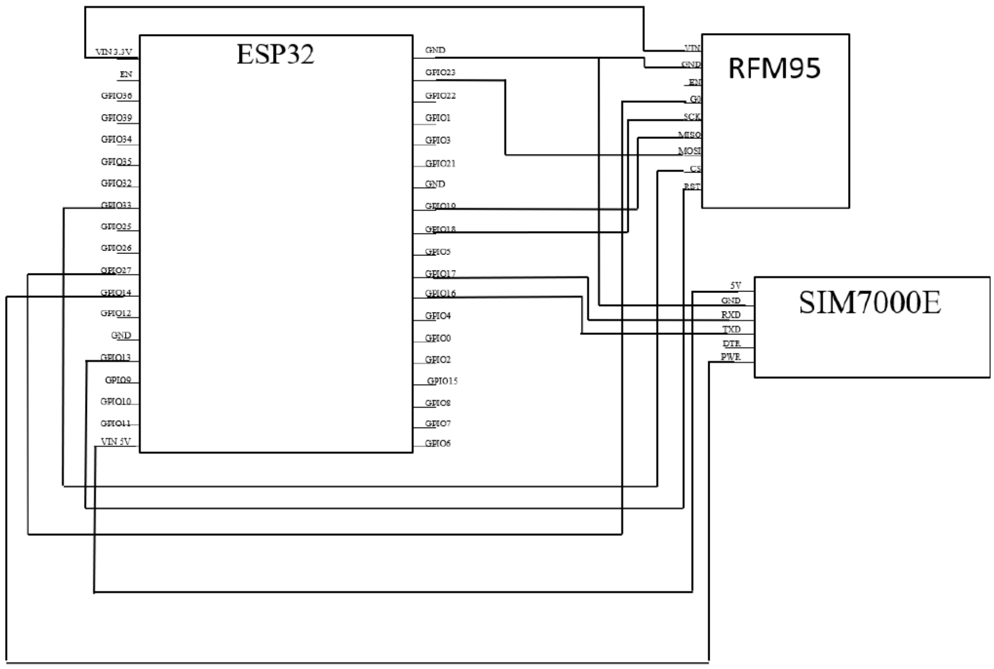


Figure 2.1. Aggregation Node Circuit

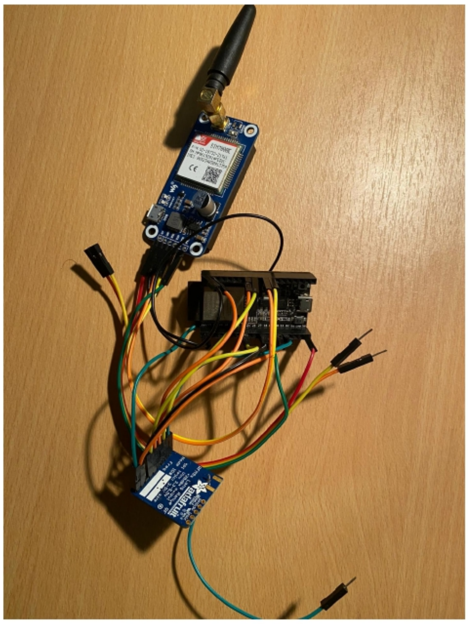

FIGURE 2.2. Aggregation Node

**2.2. Sensor Node**

Which regards to the Sensor Node, this one is the responsible to collect the necessary data in order to proceed to a correct analysis of the field, this way it needs to have in its constitution sensors of humidity and temperature of the air and the soil as well as a solar panel and a battery so that it can be installed in any area of the land without any restriction to the level of power supplying. Its build can be seen in Figure 2.3 as well as its circuit in Figure 2.4.

**2.3. Actuator Node**

In order to perform the necessary actions to keep the field in the desired conditions, the Actuator Node is equipped with water pumps in order to manage the needed water for each irrigation zone.

Besides the water pumps, this node is also equipped with a weather station (Figure 2.5) in order to collect data on wind speed and direction and rainfall, this data will contribute to an improved analysis of weather conditions. In Figure 2.6 you can see the presented node and in Figure 2.7 its circuit.

Figure 2.3. Sensor Node

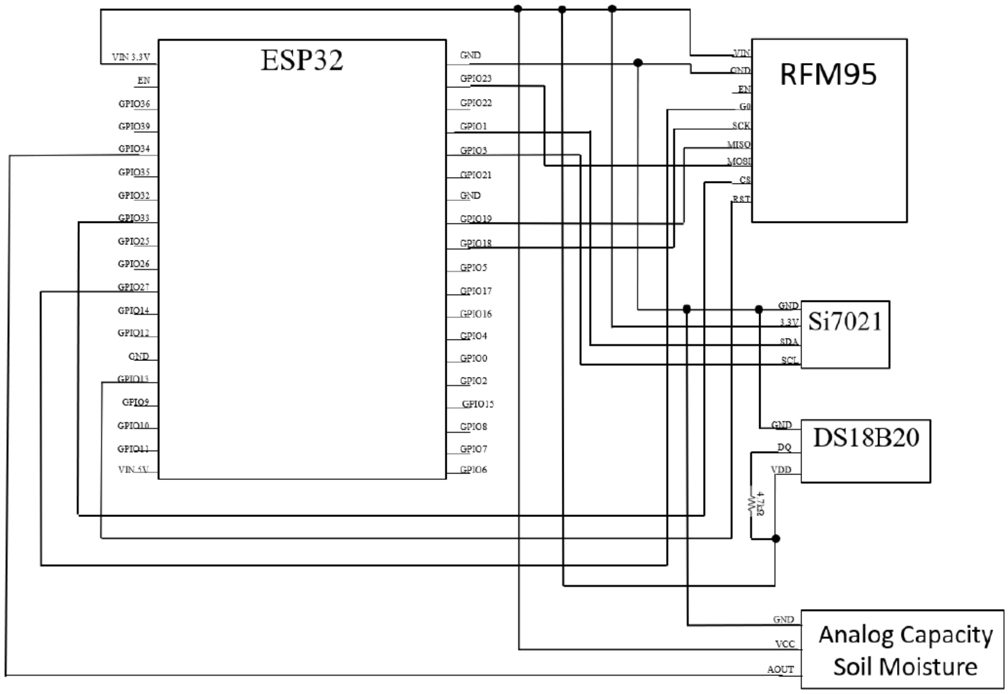


Figure 2.4. Sensor Node Circuit

Figure 2.5. Weather Station

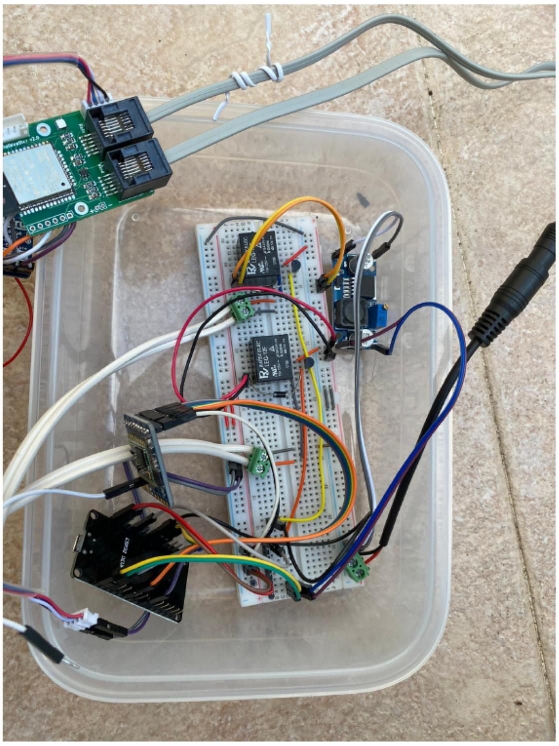

FIGURE 2.6. Actuator Node

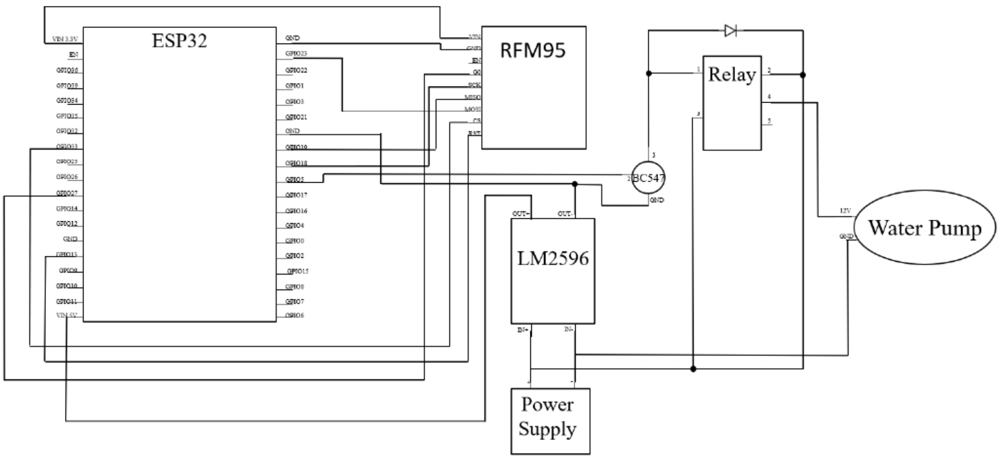


FIGURE 2.7. Actuator Node Circuit

# APPENDIX D

## Scientific Contributions

# A Methodology for Sustainable Farming Irrigation using WSN, NB-IoT and Machine Learning

João Cardoso*, André Glória*†, Pedro Sebastião*†
* ISCTE-IUL, Av. das Forças Armadas, Lisbon, Portugal
† Instituto de Telecomunicações, Av. Rovisco Pais, 1, Lisbon, Portugal
Email: jmbco1@iscte-iul.pt, afxga@iscte-iul.pt, pedro.sebastiao@iscte-iul.pt

***Abstract*—With the quick advance of technology and the appearance of low cost/high performance solutions, it became possible to develop new solutions in order to achieve sustainability. This paper proposes a scheme for monitor and control farming irrigation in order to measure and administrate the right amount of water needed to avoid overirrigation, preventing and alerting to risk situations, that require immediate intervention. This paper provides a methodology for an IoT system with a wireless sensor network installed on the ground and a main server, running machine learning algorithms, to process the collected data. Communications will be conducted using NB-IoT and LoRa. In order to give the owner the capability of consult the collected data and control his property, an iOS application completes the methodology, allowing the system to be remote and to be used anywhere, only requiring an internet connection.**

***Index Terms*—Internet of Thing, Wireless Sensor Network, NB-IoT, LoRa, iOS, ESP32, Sustainability, MQTT**

### I. Introduction

Agriculture has always been the main supplier of food for society, being responsible for more than 74% [1] of the population daily consumption. In order to keep the production for the growing number of world population, changes must take place, mainly when the growth in production also increases the consumption of water. In addition, with water scarcity being a worldwide concern, sustainability and the use of technology in agriculture have become big trends.

Until recent times, water management was done by humans, mostly with manual solutions with programmed controllers to irrigate every day.

With more than 70% of the fresh water used for agricultural irrigation related activities and according to [2], 30% of this water is being wasted or misused due to many situations like lack of control.

So, the main goal of this paper is to present a solution for monitor and control of agricultural fields. This methodology allows to establish connection between devices which are spread over the field without the need of a wired connection, turning the system more versatile.

In agriculture, the key point to success is the water administrated to the fields. It becomes necessary to monitor the fields quality to calculate the exact amount of water required and the best way and time of the day to administrate it. For that, a set of artificial intelligence algorithms will feature in the system, in order to analyze, in real time, the need of water.

When administrating the correct amount of water, not only we are saving a natural resource that nowadays is more and more in danger, but also creating a more sustainable way to create food, with better conditions and fewer cost to the farmer.

In this paper similar approaches will be analyzed in order to understand the research status and where we present innovation. After that, will be described the proposed system, including a detailed overview of each module, as well as an implementation scenario. Some experimental results will be described and finally some conclusions and future work presented.

### II. Related Work

With the advance of technologies over the years, and especially of IoT, the research for solutions to improve field monitorization and agricultural results has increased. The creation of low cost, low power consumption and ambiguous solutions are one of the most search and developed projects, however no solution was found that met all the goals of the proposed system. Within the found projects, two solutions were selected, among many other [3]–[7], and explained ahead.

In [6], the author has created a system with a specific number of nodes which contains sensors in order to collect data. The collected data is sent to a center node were, with a Web Application, is possible to consult the collected data and perform some tasks according to that. This project offers many agricultural solutions, however there are some gaps such as lack of ambiguity of the system, the center server algorithm is static for the existing nodes and the cannot be considered a low consumption solution.

The author of [8] developed a system which came with a huge number of features by using lots of sensors in order to collect data from high field dimensions. However, the developed system is very complex in terms of communications with the main server, in this case, to the cloud. Besides that, it needs a specially design configuration in order to perform the proposed task and a re-engineering of some components is needed in order to provide higher scalability.

As can been seen, there are some similar solutions compared to the presented, however these solutions do not accomplish all the proposed topics.

This work was supported in part by ISCTE - Instituto Universitário de Lisboa from Portugal under the project ISCTE-IUL-ISTA-BM-2018

978-1-7281-6445-8/20/$31.00 © 2020 IEEE

Our solution integrates innovation with the use of NB-IoT, the reduction of nodes power consumption, including self-generating power, and the creation of a solution can reduce the complexity of implementation and scale of a WSN.

### III. Proposed System Architecture

The proposed system in this paper requires an extended wireless sensor network which is controlled by nodes, being these nodes, as well as the sensors, installed on the ground. Besides the hardware implemented on the ground, it is required a main server that will receive the collected data, process it and act according to the obtained results. Finally, a mobile application is needed to consult the collected data, receive alerts and perform tasks. In this case the mobile application will be developed for iOS devices. Given this, it becomes necessary to divide the system in modules, which will be described in the next topics. An overview of the system is presented in Figure 1

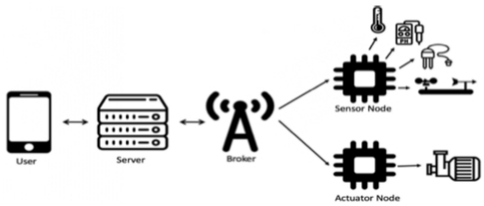


Fig. 1. System Architecture

#### A. *Communication*

Since the WSN is the main component of the system, it becomes necessary to study the best way to communicate between nodes. According to the purpose of this paper, and since the solution is to be implemented in agricultural fields, which are characterized by being in rural areas, with reduced access to energy supplies, the sensor nodes have to be powered by batteries which leads us to look for low consumption communication solutions. In order to that and according to [9] the best way to communicate between nodes is LoRa.

Besides the communication between the nodes, it becomes necessary to study the best way to send and receive data from the main server. Given that the system will be installed in agricultural areas and these are known by low connection availability, mainly in Internet connections, it becomes necessary to search for other solutions which leads us to Narrowband-IoT (NB-IoT). NB-IoT is the best option for the proposed system given his low power consumption and high capabilities of transferring data from 2G to 4G. In this particular case, the capabilities of communication in 2G are very important due to the fact that agricultural fields are known as low network coverage areas [10].

In order to transmit the data through the already spoken communication solutions, it will be used the Message Queuing Telemetry Transport (MQTT) protocol. This protocol is the one that best suits the purposed solution, given his simplicity and for being specially designed for IoT solutions [11], [12].

#### B. *Hardware*

Has shown in Figure 1, the proposed system will have different types of nodes, such as: broker, sensor nodes and actuator nodes.

Sensor node (Figure 2) are coupled with several sensors, in order to collect the needed data, and a LoRa module so it can transmit the collected data.

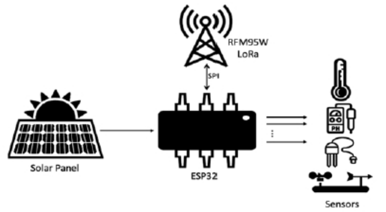


Fig. 2. Sensor Node

Actuator node (Figure 3) will manage the necessary instrument to control the field, such as water pumps to administrate water when necessary, and for that it includes a relay system. Also, a LoRa module is included, so it can receive the needed instructions to perform tasks.

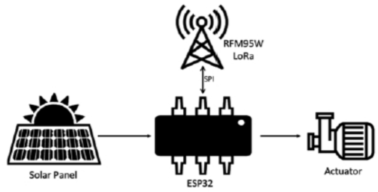


Fig. 3. Actuator Node

The broker (Figure 4) will be responsible to establish connection to sensor nodes, which will send to him the collected data, via LoRa, and then will send this data to the server, via NB-IoT. After being processed, the server will send back the instructions to execute in the actuator nodes, these last instructions will be sent to actuator nodes via LoRa.

All the nodes share the computational unit, the ESP32, which offer the less power consumption and the best process performance capabilities relation. This microcontroller which can reach consumptions as low as 10uA, due to its deep sleep capabilities, also as a dual-core processor, capable of handling the gathering, transmission and processing of data [13]

In terms of communications, as explained before, LoRa will be used to communicate between the installed nodes, and for that, the RFM95W LoRa transceiver will be used. These modules allow long-distance communication with a low

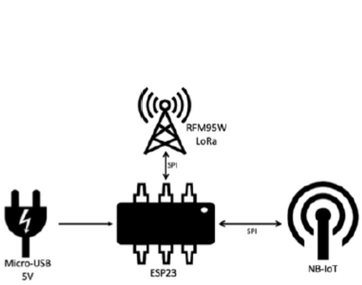


Fig. 4. Broker Node

consumption relation, being able to transmit up to 2km away with just 70mA [14].

In order to establish a connection from broker to the server, the ESP32 is coupled with a NB-IoT shield. The module that best suits the purposed system is SIM7000E, being able to establish NB-IoT connections as well as 2G connections, which becomes very important due to the lack of network coverage in some agricultural fields [10].

Finally, and since the proposed system is designed to be the most efficient possible with a low power consumption and, in order to make the project more sustainable, the power supply for the sensor nodes will be ensured by solar panels. This solution becomes even more suitable, since the fact that these sensor nodes will be installed in agricultural field which have a high sun exposure.

### C. Software

In terms of software and in order to become possible to process the collected data for the system to be able to act according to that, in the server some algorithms will be running.

These algorithms will calculate the amount of water needed for a specific zone, depending on the type of crops, the type of valves and tubbing used, the distance between these same valves and the number of irrigations in one day. Considering all these parameters, and using the equation shown (Equation 1), it becomes possible to calculate the amount of water that the particular garden in analysis need.

$$T = \frac{A \times (K_c + ET) \times 60}{F \times N \times 1000} / P \tag{1}$$

In the equation 1, the result, *T*, represent the irrigation time in minutes, *Kc* is crop coefficient, *F* is the outgoing flow of water in m3/h, *N* is the number of valves, *P* is the number of irrigation periods and *A* is the garden area in m2, this last is given by equation 2.

$$A = [(0.5 \times N) - 1] \times D^2 \tag{2}$$

In equation 2, *D* represents the distance between valves

$$ET = 0.0023 \times (T_{med} + 17.78) R_0 \times (T_{max} - T_{min})^{0.5} \tag{3}$$

In equation 3 it is possible to see the simplified Hargreaves formula. This one will give the evapotranspiration ($ET$) in mm/day, where *Tmed* is the average temperature, *Tmax* the maximum temperature, *Tmin* is the minimum temperature and *R0* is the incident extra-terrestrial solar radiation [15].

In order to obtain the best results possible, in a second stage will be implemented Machine Learning algorithms to predict the values variation in order to obtain the most precise results possible.

Besides the data processing, it is also necessary that the collected data arrives to server. For that each node needs to have a script capable of establishing communications between the installed nodes through LoRa. For that, the RadioHead library will be used [16]. This library allows to give addresses to each node and, through theses addresses, became possible to specify the target of each message, avoiding unnecessary broadcasts. RadioHead brings the possibility of checking if the messages are correctly sent and received through messages acknowledgments (ACK), avoiding message lost.

### D. Visualization and Control

In terms of visualization, and to become possible to control the field, from consulting to acting according to the collected data, an iOS application is given to the user, so that he can consult the status of their field anytime, anywhere, needing only an internet connection on his smartphone.

This application is connected to the server and to the broker, allowing it to check the data base stored on the server and, at the same time, act on the actuators, such as valve to administrate water at a specific irrigation zone, sending a message, via MQTT, to broker.

An important feature of the mobile application is the possibility to alert the owner for any situation that should not occur. In this specific case, the algorithms running on server will detect an unexpected value and will send an alert to the user. This alert will appear in a notification format. In Figure 5 can be seen a flowchart of the proposed algorithm.

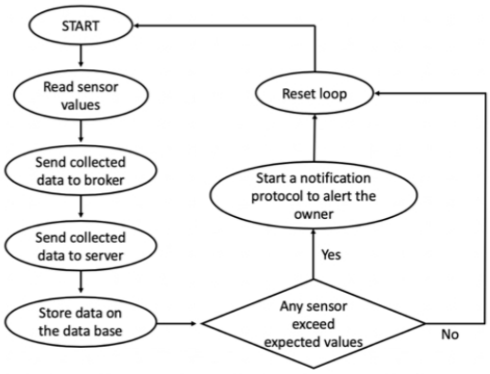


Fig. 5. System Flowchart

## IV. Proposed System Implementation

The main goal of this paper is to present a new methodology to monitor and control agricultural fields, in order to improve irrigation efficiency.

Portugal is well known for its agricultural regions, mainly for the vineyard fields, and being grapes a culture that need a precise control on the amount of water used, it is a great use case for our solution.

Being the system composed by wireless sensor nodes that individually retrieve data from the local implementation, multiple nodes need to be implemented in order to truly cover the entire field. As shown in Figure 6, each sensor needs to be placed in a vineyard row, with a recommend distance of 5 meters between nodes. This allows for a more precise analysis of the required parameters, with an average of 50 sensors nodes per hectare.

Fig. 6. Sensor Node Distribution

Each of the sensor nodes, as stated before, is coupled with a set of sensors. In order to achieve the goal of understanding the ideal irrigation times these sensors need to collect information's that can help analyze the need of water. For that each node can include the following:

- DS18B20, a waterproof soil temperature waterproof, capable of collect data between - 55ºC and 125ºC with an accuracy of 0.5ºC for -10ºC to 85ºC interval. This is a simple sensor with an easy implementation, only requiring a one wire connection [17].
- SI7021, a temperature humidity sensor, capable of collect temperature and moisture values with high precision (± 3%RH, ±4ºC). About range features, this sensor can collect values between -40ºC and 125ºC in terms of temperature and 0 to 100% in terms of moisture values [18].
- Sen0249, a soil pH sensor which can be directed stab into semisolid material with high precision [19].
- Analog Capacity Soil Moisture, a humidity waterproof sensor which provides the capacity of collecting data with high precision. This collection is made by capacitive sensing [20].

Besides the sensors spread throughout the field connected directly to the nodes, also a weather station is installed. The SEN 1086, a Weather Station which provides the capability of collect environmental data on the field, has the ability to collect temperature values between -40º and 80ºC, humidity values between 0 and 99%, and is equipped with an anemometer, used to measure wind speed, a wind vane, for collection of wind direction and a rain bucket, in order to collect precipitation values [21].

The weather station will have a dedicated node, since the values it collects are ambiguous for the entire field, so no more than one need to be installed in the field.

Besides the sensors, also a set of actuator nodes will be installed, one for each irrigation pump. Since vineyard are usually irrigated with a drip-by-drip technique, only a pump for each line is needed to control.

Finally, one broker is installed on a strategical location, in order to be able to control and communicate with every node.

All the proposed system is ambiguous and flexible, making possible to add more nodes later to turn it more efficient, not compromising the network capabilities, application features or the written algorithms.

## V. Experimental Results

Before it can be installed in a real agricultural field, some laboratorial tests were performed to check the efficiency, reliability and performance of the proposed methodology.

Firstly, each individual node was assembled, as can be seen in Figures 7, 10 and , where appears one sensor node, the actuator node connected to a pump, and the broker node with the NB-IoT shield.

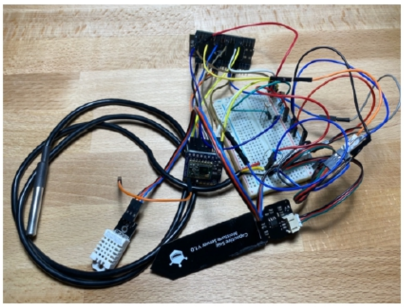

Fig. 7. Sensor Node

To test the data acquisition and transmission, one sensor node was installed in a similar environment, in this case a domestic garden, where it was left running for a couple of days retrieving data each 15 minutes and sending them to the broker, installed inside the house.

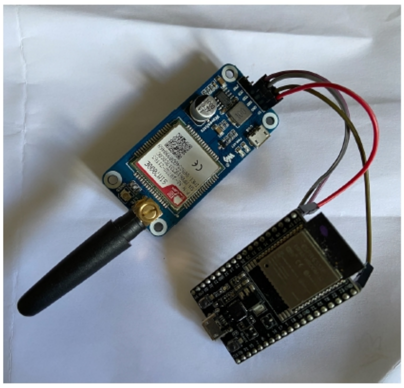

Fig. 8. Broker node

Some of the gathered data can be seen in Figure 9, showing also the developed iOS mobile application.

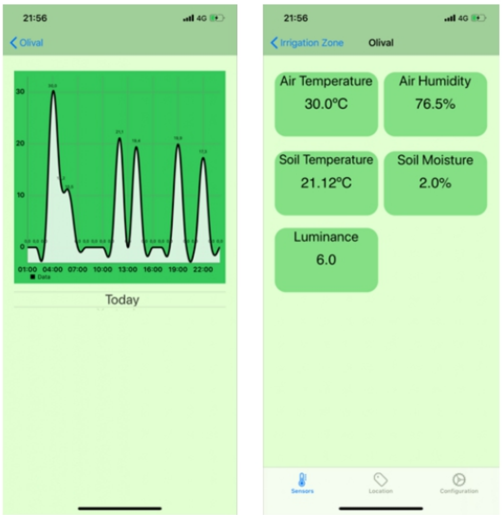


Fig. 9. iOS application

To test the actuator nodes, it was connected to a 24V pump, as seen in Figure 10, which is turned on via the iOS application.

Finally, and since energy is one the most critical parts of this system, the sensor node was connected to a digital multimeter in order to check the energy used in its lifecycle, in order to better fit the solar panel. Table I presents the obtained results.

The algorithms were also tested, using the data retrieved from the sensor node, in order to check if they can predict the ideal irrigation times. The results were not conclusive, since a small dataset was used, so more adjusts and training needs to be done, before the final implementation.

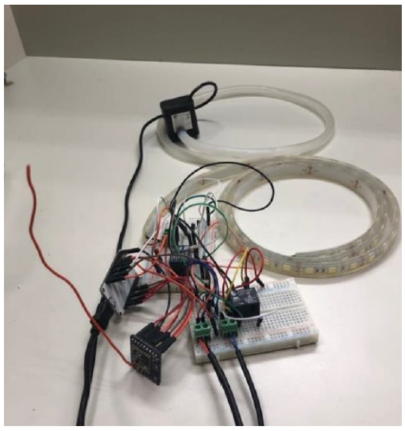

Fig. 10. Actuator Node with pump

TABLE I
SENSOR NODE CONSUMPTION'S

| Node Status | Power Consumption $V_{IN}$ = 3V |
|---|---|
| Transmitting | 80 mA |
| Collecting data | 30 mA |
| Deep Sleep | 100 $\mu$A |

## VI. CONCLUSIONS AND FUTURE WORK

This paper proposes a system that provides agricultural field owners the capabilities of control his properties anytime and anywhere, being alerted for any risk situation that occur in his absence. All of this targeted for a low-cost solution with the purpose of saving natural resources using new technologies such as NB-IoT and exploring the ever-increasing capabilities of IoT. Besides the communication solutions explored, the solar panel options are also an even more explorable feature in order to get the most sustainable solution possible.

With the experimental tests done it is possible to conclude that the methodology can be applied in a real case scenario, although some modification, mainly in the software part, are needed. Also, based on the energy consumptions of the sensors nodes, a solar power battery can be developed, since the nodes has low consumptions.

The next step, after concluding the previous stated tasks, is to implement the all system in a real agricultural field and implement a machine learning solution in order to get the best predictions possible. Besides the implementation on agricultural fields, the proposed system may also be implemented in public or private gardens.

REFERENCES

[1] Food and Agriculture Organization of the United Nations, “Water for Sustainable Food and Agriculture,” 2019. [Online] Available: http://www.fao.org/3/a-i7959e.pdf, (visited 10/01/2020).

[2] U. S. E. P. Agency, “Why save water? Statistics and Facts.” [Online] Available: ttps://www.epa.gov/watersense/statistics-and-facts, (visited 10/05/2020).

[3] M. K. Gayatri, J. Jayasakthi, and G. S. Mala, “Providing Smart Agricultural solutions to farmers for better yielding using IoT,” *Proceedings - 2015 IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development, TIAR 2015*, no. Tiar, pp. 40–43, 2015.

[4] R. C. Andrew, R. Malekian, and D. C. Bogatinoska, “IoT solutions for precision agriculture,” *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, pp. 345–349, 2018.

[5] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, “A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming,” *IEEE Access*, vol. 7, pp. 156237–156271, 2019.

[6] A. Mondal, I. S. Misra, and S. Bose, “Building a low cost solution using wireless sensor network for agriculture application,” *Proceedings of 2017 International Conference on Innovations in Electronics, Signal Processing and Communication, IESC 2017*, pp. 61–65, 2017.

[7] R. Dagar, S. Som, and S. K. Khatri, “Smart Farming - IoT in Agriculture,” *Proceedings of the International Conference on Inventive Research in Computing Applications, ICIRCA 2018*, no. Icirca, pp. 1052–1056, 2018.

[8] C. Kamienski, J. P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T. S. Cinotti, R. F. Maia, and A. T. Neto, “Smart water management platform: IoT-based precision irrigation for agriculture,” *Sensors (Switzerland)*, vol. 19, no. 2, 2019.

[9] A. Gloria, F. Cercas, and N. Souto, “Comparison of communication protocols for low cost Internet of Things devices,” *South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference, SEEDA-CECNSM 2017*, 2017.

[10] L. Feltrin, M. Condoluci, T. Mahmoodi, M. Dohler, and R. Verdone, “NB-IoT: Performance estimation and optimal configuration,” *24th European Wireless 2018 "Wireless Futures in the Era of Network Programmability", EW 2018*, pp. 50–55, 2018.

[11] D. Ghosh, A. Agrawal, N. Prakash, and P. Goyal, “Smart saline level monitoring system using ESP32 and MQTT-S,” *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services, Healthcom 2018*, pp. 1–5, 2018.

[12] Bharati Wukkadada ; Kirti Wankhede ; Ramith Nambiar ; Amala Nair, “Comparison with HTTP and MQTT In Internet of Things (IoT) - IEEE Conference Publication,” *2018 International Conference on Inventive Research in Computing Applications (ICIRCA 2018)*, no. Icirca, pp. 249–253, 2018.

[13] Espressif, “ESP32 Series Datasheet,” 2019. [Online] Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, (visited 10/05/2020).

[14] Adafruit, “RFM95W LoRa Radio Transceiver.” [Online] Available: https://www.adafruit.com/product/3072, (visited 10/01/2020).

[15] A. Glória, P. Sebastião, C. Dionísio, G. Simões, and J. Cardoso, “Water management for sustainable irrigation systems using internet-of-things,” *Sensors (Switzerland)*, vol. 20, no. 5, pp. 1–14, 2020.

[16] Adafruit, “Adafruit RadioHead.” [Online] Available: https://github.com/adafruit/RadioHead, (visited 10/05/2020).

[17] Maximintegrated, “DS18B20 Programmable Resolution 1-Wire Digital Thermometer DS18B20 Programmable Resolution 1-Wire Digital Thermometer Absolute Maximum Ratings,” vol. 92, pp. 1–20, 2019.

[18] S. Labs, “Si7021-A20 Datasheet,” pp. 1–36, 2016. [Online] Available: https://www.silabs.com/documents/public/data-sheets/Si7021-A20.pdf, (visited 10/05/2020).

[19] DFRobot, “Gravity: Analog Spear Tip pH Sensor / Meter Kit (For Soil And Food Applications) SKU: SEN0249,” 2017. [Online] Available: https://wiki.dfrobot.com/Gravity__Analog_Spear_Tip_pH_Sensor___Meter_Kit__For_Soil_And_Food_Applications__SKU__SEN0249, (visited 10/05/2020).

[20] DFRobot, “Capacity Soil Moisture Sensor.” [Online] Available: https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193, (visited 10/05/2020).

[21] DFRobot, “Wheather Station.” [Online] Available: https://wiki.dfrobot.com/Weather_Station_with_Anemometer_Wind_vane_Rain_bucket_SKU_SEN0186, (visited 06/01/2020).

# Improve Irrigation Timing Decision for Agriculture using Real Time Data and Machine Learning

João Cardoso
Department of Science,
Technology and Information
ISCTE - Instituto
Universitário de Lisboa
Lisbon, Portugal
Email: jmbco1@iscte-iul.pt

André Glória
Department of Science,
Technology and Information
ISCTE - Instituto
Universitário de Lisboa
Lisbon, Portugal
Email: afxga@iscte-iul.pt

Pedro Sebastião
Department of Science,
Technology and Information
ISCTE - Instituto
Universitário de Lisboa
Lisbon, Portugal
Email: pedro.sebastiao@iscte-iul.pt

***Abstract*—With the constant evolution of technology and the constant appearance of new solutions that, when combined, manage to achieve sustainability, the exploration of these systems is increasingly a path to take. This paper presents a study of machine learning algorithms with the objective of predicting the most suitable time of day for water administration to an agricultural field. With the use of a high amount of data previously collected through a Wireless Sensors Network (WSN) spread in an agricultural field it becomes possible to explore technologies that allow to predict the best time for water management in order to eliminate the scheduled irrigation that often leads to the waste of water being the main objective of the system to save this same natural resource.**

***Index Terms*—Machine Learning, Neural Network, Decision Tree, Support Vector Machine, XGBoost, Random Forest, Sustainability, Smart Irrigation**

## I. Introduction

Agriculture has always been the main supplier of food for society, being responsible for more than 74% [1] of the population daily consumption. In order to keep these production values and evolving them to meet the needs of the increasing population, water consumption has been the main concern. In this topic, resources to technological solutions have demonstrate to be the best bet to meet the needs.

Regarding the issue of water consumption, its application continues to be mostly controlled by human means, leading many times to mismanagement, higher than necessary consumption and consequently to higher financial expenses and the waste of this natural resource.

Technological solutions have been increasingly concerned with monitoring the fields in order to understand the best times for their harvest and controlling their values throughout the season. This fact makes possible a more in-depth research in order to predict the best water management heights, leading to less waste and an improvement in harvests.

Thus, the main objective of this paper is to understand the possibility to predict, automatically, using machine learning solutions, the best time of day for irrigation, based on local sensor and weather data.

This work was supported in part by ISCTE - Instituto Universitário de Lisboa from Portugal under the project ISCTE-IUL-ISTA-BM-2018

So, and based on related work in the area of agricultural land monitoring, it is possible to make the correct analysis of which data are the most important to take into account when developing the algorithms [2].

The system already implemented by the authors in [2] will be the basis of the research for this paper. This system uses a wide range of sensors that are strategically spread over the agricultural fields in order to collect the data needed for correct monitoring. Then this data is sent, using Wireless Sensor Network (WSN), to a central server, where a database is hosted, in order to be able to store the data collected over long periods of time. This high number of data will allow the correct development and training of the algorithms to accomplish the proposed goal of this paper, understating which is the best machine learning algorithm to predict the ideal irrigation hours.

According to the results obtained by the authors in [3], it is possible to reach savings up to 40% in water. These water saving results are obtained only through the study of formulas that calculate the amount of water that needs to be administered to the fields. So, the implementation of machine learning algorithms, as this paper will study, is in a good position to achieve even better savings results of this natural resource.

This article presents a study of Machine Learning Classification algorithms where Random Forest (RF), Neural Networks (NN), XGBoost, Decision Trees (DT) and Support Vector Machine (SVM) will be studied.

In addition, and in order to be able to study correctly the algorithms already mentioned, a dataset and a methodology were created in order to enable the correct study, leading to the best possible optimization. The whole process of creating the spoken dataset and the methodology followed will be explained later. Later will be presented the conclusions drawn from the study present in this paper.

## II. Related Work

With the evolution of technology and the constant development of solutions in the area of the Internet of Things (IoT) in parallel with intelligent solutions produced in the area of artificial intelligence and machine learning, multiple solutions have already been developed with the aim of combining the two in order to obtain cheaper solutions and with the purpose of saving natural resources.

In the study made in [4], the author developed a system that applies artificial neural network techniques "for water level prediction, a fuzzy logic control algorithm for sluice gate setting period estimation, and hydraulics equations for sluice gate level adjusting". As input for the pretended prediction, the author only gives to the algorithm a dataset with the last three days of water level, being this a small amount of data since the main goal is to reach the most exact prediction possible being the amount of data given as input a concern.

Regarding to the study made by the author of [5], this article presents a study of machine learning applications in agricultural supply chains. Although the author have not developed any system or script, had conclude, through a heavy research that the most explored algorithms are neural Networks, being these the most used for agricultural solutions.

In [6], the authors present, through the collection of land data using a WSN, a study where machine learning algorithms (SVM and RF) are applied in order to understand the irrigation needs of the land under study, with an accuracy in the order of 80%. Although the study developed in [6] has an intensive research on the level of collected data, and there has been an investment in the analysis of the values through formulas that allow the calculation of the necessary water values, with regard to the algorithms used, the whole Machine Learning solution was based on previous research on the algorithms and the development of the dataset. This may lead to a poor adaptation of the developed technology to the solution where it will be applied.

## III. Dataset & Data Pre-processing

In order to develop the correct algorithm that can predict the best time of day for water administration, it is necessary to compose a correct and useful dataset.

The dataset used, as said before, was created from a range of data previously collected through the implementation of the system studied in [2]. Furthermore, the data were complemented by values provided by the Portuguese Sea and Atmosphere Institute (IPMA).

This data contains a vast number of features, as can be seen in Table I. All entries were individually analyzed, and extra features were added to each one, making the dataset richer and more substantiated in order to facilitate the process of training the algorithms under study and achieving better results.

Within these are the features *"Is_favorable"*, which classifies the ground conditions at that specific time as favorable or unfavorable for sustainable irrigation, *"Need_Irrigation"* which indicates if irrigation is needed and *"Had_Irrigation"* where it is indicated if the land has already been watered. These values were calculated based on the sensor data collected. Finally each entry was manually classified with the best irrigation hour according to the real time sensor data, within the label *"Suggested_Hour"*.

After conclusion, the dataset used has 105217 entries.

TABLE I
DATASET PROPERTIES

| Feature | Description |
|---|---|
| Year | Timestamp Year |
| Month | Timestamp Month |
| Day | Timestamp Day |
| Hour | Timestamp Hour |
| Temperature | Air Temperature [ºC] |
| Relative_Humidity | Air Humidity [%] |
| Total_Precipitation_Low | Precipitation [mm/H] |
| Wind_Speed | Wind Speed [km/h] |
| Wind_Direction | Wind Direction [º] |
| Soil_Humidity | Soil Moisture [%] |
| Had_irrigation | Field irrigated [0/1] |
| Need_Irrigation | Field needs irrigation [0/1] |
| Is_Favorable | Conditions favorable for irrigation [0/1] |
| Suggested_Hour | Suggested irrigation hour |

## IV. Machine Learning Classification Algorithms

Classification algorithms are the chosen ones to use in this study. Classification is the process of predicting decision values in the qualitative or category class of a given data point.

Random Forest (RF) is a tree-based method that conglomerates several self-determining decision trees developed for classification and regression. Through the combination of the various trees it is able to understand which is the best option, being the main objective to reach one in pure i.e, a node formed by a single class, giving it high predictive capabilities [7].

Decision Trees (DT) are tree based methods in which each path begin in a root node representing a sequence of data divisions until reach a Boolean outcome at a leaf node. These methods can be applied for classification and regression. The final goal of this method is to reach a model that can predict the search value for that specific scenario by learning simple decision rules [8].

Support Vector Machines (SVM) are a set of supervised learning methods developed for classification, regression and outlier's detection which is known by his high effective in high dimension spaces and for its use for training points in the decision function, being also memory efficient [9].

For the study of Neural Networks algorithms, which are defined as computational models of nervous system, the Multilayer Perceptron (MLP) method was used, which is a supervised learning method that learns a function $f(.) : R^m \rightarrow R^o$ by training on a data set, where $m$ is the number of dimensions to input and $o$ the number of dimensions for output. These MLP networks are characterized by being general-purpose, flexible and non-linear. Their complexity can be changed according to their application by varying the number of layers and units of each layer [10], [11].

Regarding XGBoost, this is a boosting tree based method which in turn are based on decision trees. Considering that the linear combination for multiple trees capabilities that can well fit the training data and describe the complex non linear relationship between input and output data, makes this method considered one of the best methods in statistical learning [12].

## V. Methodology

The methodology followed in the development and improvement of the algorithms previously described was divided into 4 phases.

In a first phase the dataset was built, following the steps previously described. After the dataset was completed, all values were analyzed and each entry classified as favorable or unfavorable for water administration, whether or not it needs water, if the land has already been watered and the best hour to irrigate (*"Is_favorable", "Need_Irrigation", "Had_Irrigation" and "Suggested_Hour"* respectively).

In a second phase, with the completed dataset, and before testing the various algorithms, a test was made on the importance of each feature of the dataset, allowing to understand which are the most important and which should not be considered at the time of training, in order to optimize the dataset and leading to the elimination of noise.

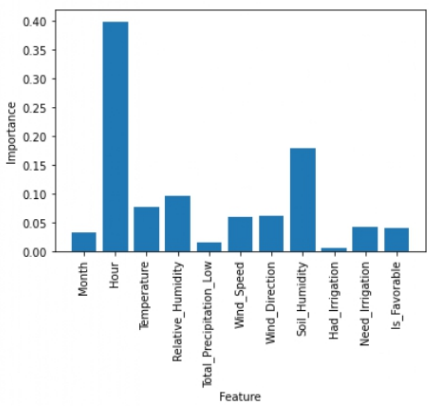


Fig. 1. Feature importance

As can be seen in Figure 1, the features *"Day", "Need_Irrigation", "Month", "Total_Precipitation_Low" and "Had_Irrigation"* have a low importance for the training and later result of the algorithms, so they were discarded. This discarding also results in a shorter time in the training of the algorithms and no significant variation was observed in the results obtained after the discarding.

The third phase of the methodology consisted in training the five algorithms under study using the parameters of each one with default values. In order to study and improve these algorithms, scikit-learn was used. This is an open source Machine Learning library developed for Python implementation [13]. For the implementation of XGBoost, a library made available by this same algorithm was used. It implements machine learning algorithms under the Gradient Boosting framework [14]. In this phase the goal is to understand which algorithms have better results using the dataset to predict the best irrigation time.

In the fourth and last phase, the best algorithms from the previous phase were exhaustively tested in order to understand what would achieve better accuracy values. In these tests an hyperparametrization tuning was done to each algorithm, in order to understand the best scenario possible. For this, a method provided by scikit-learn called RandomizedSearchCV was used, which performs the fit and training of the algorithm under study, calculating which parameters are best suited to it [15].

The final algorithm is then adapted to be running on the central server of the system under test, where the database is hosted and where the values will be received in real time. Thus, the algorithm receives the collected values and calculates the best time for water administration. When crossing the resulting values of the algorithm developed with the algorithms previously spoken and already developed by [2] that calculate the amount needed to administer the terrain under analysis, it will lead to a better management an to higher water saving values, being this the main objective.

Figure 2 shows the flowchart of the described methodology.

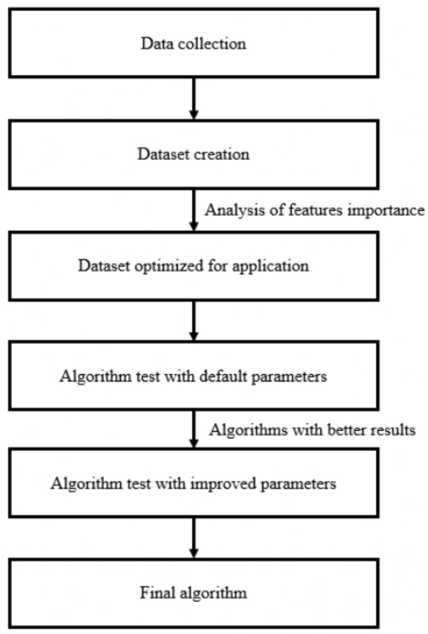


Fig. 2. Methodology Flowchart

## VI. Results & Discussions

In order to put into practice the described methodology, it is necessary to precede the training and subsequent testing of the various algorithms under study to understand what best suits the intended application.

To check if the algorithms have a good applicability when receiving real values, the dataset was divided in two parts. One with 70% of the dataset, that will be used to train the algorithm, and other with the remaining 30%, will be used to test the accuracy of the trained models.

Table II shows the default parameters used for each of algorithms.

The accuracy results obtained for each model can be seen in Table III. As can be seen, the accuracy values of each algorithm trained were quite varied, which allowed to see which were best suited to the dataset and application under study.

It is possible to notice that the SVM does not fit in the proposed goal, having a low accuracy value and, even with the variation of some parameters, it would not be possible to achieve acceptable values. As for the other algorithms, although higher when compared to SVM, it is possible to conclude that only two stand out even before any optimization. As such XGBoost and RF will be further evaluated, since DT and NN, even after optimized, will not be able to reach better accuracy.

Moving forward to the next training phase, using only XGBoost and RF, and based on the parameters shown in Table II and the documentation for each of these algorithms, the hyperparametrization tuning was made only for the most important parameters, mainly those who have a numerical value. Table IV shows the tuning options for each of the selected algorithms.

The results obtained with the hyperparametrization tuning, including the best parameters settings and accuracy, can be observed in Table V. Through the analysis of the results, after the optimization of the various parameters of the two algorithms under study it is then possible to conclude that by choosing the best parameters, instead of the default configuration, it is possible to improve the accuracy of the models. Although is is not a huge improvement, 1% for XGBoost and only 0.1% for RF, this improvement can lead to the saving of a huge amount of water.

In terms of the algorithm that has the best accuracy for the situation and dataset tested is the XGBoost, which will be used for the solution.

## VII. Conclusion

In this article a study of machine learning algorithms was made in order to understand which will have the higher accuracy when classifying the ideal hour to irrigate an agricultural field, based on local sensor and weather data. The algorithms tested included Random Forest, Neural Network, XGBoost, Decision Trees and Support Vector Machine.

The literature on this topic showed that research is already being done to calculate the amount of water to be administered to the agricultural field, however the time of day at which this administration was done continues to be decided by the owner and in a poorly founded way.

A methodology was followed to obtain a suitable dataset for the study and several scenarios were explored in order to understand which algorithm best suited the situation under study, and it was concluded that XGBoost was the most suitable.

After the optimization of the tested algorithm it was possible to reach an accuracy in the order of 87%, which leads to believe that the final result can improve water management and consequent savings of this natural resource.

Comparing to the results obtained by [6], with 80% accuracy using RF, our methodology obtains better results with XGBoost. Also, in terms of comparison, when using RF, our methodology also gets better results, with 84% accuracy.

As future work for this study is included the implementation of the algorithm developed in a real situation in order to test the water saving values that can be achieved and also the attempt to optimize even further the system. In order to be even more effective, and since the proposed algorithm predicts the best hour of the day for irrigation, the developed algorithm should be implemented in parallel with algorithms that calculates the amount of water needed to manage the land under study. All of these implementations should have in mind the collection of data in real time, leading to a quick response for any type of situations.

## References


[1] Food and Agriculture Organization of the United Nations, "Water for Sustainable Food and Agriculture," 2019. [Online] Available: http://www.fao.org/3/a-i7959e.pdf, (visited 20/08/2020).

[2] J. Cardoso, A. Glória, and P. Sebastião, "A Methodology for Sustainable Farming Irrigation using WSN, NB-IoT and Machine Learning," in *5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA20)*, 2020.

[3] A. Glória, P. Sebastião, C. Dionísio, G. Simões, and J. Cardoso, "Water management for sustainable irrigation systems using internet-of-things," *Sensors (Switzerland)*, vol. 20, 3 2020.

[4] B. Suntaranont, S. Aramkul, M. Kaewmoracharoen, and P. Champrasert, "Water irrigation decision support system for practicalweir adjustment using artificial intelligence and machine learning techniques," *Sustainability (Switzerland)*, vol. 12, 3 2020.

[5] R. Sharma, S. S. Kamble, A. Gunasekaran, V. Kumar, and A. Kumar, "A systematic literature review on machine learning applications for sustainable agriculture supply chain performance," *Computers and Operations Research*, vol. 119, 7 2020.

[6] A. Vij, S. Vijendra, A. Jain, S. Bajaj, A. Bassi, and A. Sharma, "IoT and Machine Learning Approaches for Automation of Farm Irrigation System," in *Procedia Computer Science*, vol. 167, pp. 1250–1257, Elsevier B.V., 2020.

[7] M. Mamdouh, M. A. Elrukhsi, and A. Khattab, "Securing the Internet of Things and Wireless Sensor Networks via Machine Learning: A Survey," in *2018 International Conference on Computer and Applications, ICCA 2018*, 2018.

[8] F. J. Yang, "An extended idea about decision trees," in *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*, pp. 349–354, Institute of Electrical and Electronics Engineers Inc., 12 2019.

[9] Q. Tang, X. Ge, and Y. C. Liu, "Performance analysis of two different SVM-based field-oriented control schemes for eight-switch three-phase inverter-fed induction motor drives," in *2016 IEEE 8th International Power Electronics and Motion Control Conference, IPEMC-ECCE Asia 2016*, 2016.