escola superior de tecnologia e gestão
instituto politécnico de leiria

Proposal of architecture for IoT solution for monitoring and management of plantations

Master degree in Computer Engineering – Mobile Computing

Emerson de Moraes Navarro

Leiria, November of 2020

escola superior de tecnologia e gestão
instituto politécnico de leiria

Proposal of architecture for IoT solution for monitoring and management of plantations

Master degree in Computer Engineering – Mobile Computing

Emerson de Moraes Navarro

Dissertation under the supervision of Professor Dr. António Manuel de Jesus Pereira.

Leiria, November of 2020

Originality and Copyright

This dissertation report is original, made only for this purpose, and all authors whose studies and publications were used to complete it are duly acknowledged.

Partial reproduction of this document is authorized, provided that the Author is explicitly mentioned, as well as the study cycle, i.e., Master degree in Computer Engineering - Mobile Computing, 2018/2020 academic year, of the School of Technology and Management of the Polytechnic Institute of Leiria, and the date of the public presentation of this work.

Dedication

This dissertation is dedicated to my wife Cristina who encouraged me to cross an ocean to begin this master's degree and supported me throughout this journey.

Thank you, none of this would have been possible without you.

Acknowledgments

I would like to express my gratitude to my primary supervisor, Prof. Dr. António Pereira, who guided me throughout this project.

I would also like to thank Professor Nuno Costa for his contribution during the publication of the article.

To my family and friends for their presence and support in the most complicated moments along this journey.

My brother Daniel, for his valuable insights and reviews, which greatly contributed to the success of this work.

Finally, I especially thank my wife Cristina, for her incentive to start this study, and for her constant patience and support throughout this period, as well as for her contribution in reviewing this work.

Introductory Note

The work presented in this dissertation was carried out in the School of Technology and Management of the Polytechnic Institute of Leiria and resulted in the following publication:

E. Navarro, N. Costa, and A. Pereira, “A Systematic Review of IoT Solutions for Smart Farming,” Sensors, vol. 20, no. 15, p. 4231, Jul. 2020, doi: 10.3390/s20154231.

Abstract

The world population growth is increasing the demand for food production. Furthermore, the reduction of the workforce in rural areas and the increase in production costs are challenges for food production nowadays. Smart farming is a farm management concept that may use Internet of Things (IoT) to overcome the current challenges of food production This work presents a systematic review of the existing literature on smart farming with IoT. The systematic review reveals an evolution in the way data are processed by IoT solutions in recent years. Traditional approaches mostly used data in a reactive manner. In contrast, recent approaches allowed the use of data to prevent crop problems and to improve the accuracy of crop diagnosis. Based on the finds of the systematic review, this work proposes an architecture of an IoT solution that enables monitoring and management of crops in real time. The proposed architecture allows the usage of big data and machine learning to process the collected data. A prototype is implemented to validate the operation of the proposed architecture and a security risk assessment of the implemented prototype is carried out. The implemented prototype successfully validates the proposed architecture. The architecture presented in this work allows the implementation of IoT solutions in different scenarios of farming, such as indoor and outdoor.

Keywords: IoT technologies, Smart farming, Architecture, Big Data, Cloud computing

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

ACM	Association of Computing Machinery
AI	Artificial Intelligence
API	Application Programming Interface
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CoAP	Constrained Application Protocol
GPIO	General Purpose Input Output
GPRS	General Packet Radio Service
GPS	Global Positioning Systems
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
JSON	Javascript Object Notation
LAI	Leaf Area Index
LED	Light-Emitting Diode
LPWAN	Low Power Wide Area Networks
M2M	Machine-to-Machine
MQTT	Message Queueing Telemetry Transport
NFC	Near Field Communication
NoSQL	Not Only Structured Query Language
OS	Operating System
POWER	Prediction of Worldwide Energy Resources
PRISMA	Preferred Reporting Items for Systematic Reviews
REST-API	Representational State Transfer
RFID	Radio-Frequency Identification
RF-ISM	Radio Frequency – Industrial Scientific and Medical
SBC	Single-Board Computers
SoC	System-on-a-Chip

TCP/IP	Tranmission Control Protocol/ Internet Protocol
UAV	Unmanned Aerial Vehicles
UDP	User Datagram Protocol
WSN	Wireless Sensor Networks

1. Introduction

The challenge of food production in the 21st century is an increasingly relevant theme as population growth increases year after year. It is estimated that by 2050 the world will have between 9.4 and 10.1 billion people who depend on the world’s biodiversity to live, increasing the demand for dedicated food production areas – specifically for planting and livestock [1]. Environmental changes caused by human beings could potentially cause conditions in which the development of new crops is not possible. Likewise, the growing urbanization decreases labor in areas typically involved in food production, increases costs and reduces the productive capacity of the sector [2].

In face of this, it becomes evident the need for the use of techniques and technologies capable of responding to the demands of the population and, at the same time, facing the challenges inherent in the reduction of labor in rural areas. The use of technology applied to agriculture is a common practice that contributes to a new concept denominated smart farming [3]. Thus, smart farming is associated with the incorporation of information systems and communication technologies to agricultural production equipment and machinery, such as agricultural information management systems, use of sensors, data analysis, global positioning systems (GPS), and communication networks [4]. These information systems and communication technologies may be applied to several applications in the agricultural context, such as management and tracking of agricultural machinery [5], [6], monitoring of silos, monitoring of water resource, and fuel [7]–[10], as well as enabling the collection of a multitude of information from crops (e.g., climate data, fertilizer, soil and plant health) [11]–[14].

The systematic use of information systems and communication technologies at the various levels and scales of the agricultural production enables a better decision making, allowing actions to be executed at the right time, quantity and location, leveraging productivity and minimizing waste [15]. A particularly relevant concept within this scenario is the Internet of Things (IoT). IoT has the capacity to instrumentalize producers, giving greater visibility to important cultivation information during all phases of food production - from planting to product distribution - and, thus, access to data that support the decision-making [16]. Given the relevance of IoT, the use of IoT in agriculture has been promoted by governments of the world's largest agricultural producers, such as Brazil and the European Union, through policies, incentive programs for the incorporation of new technologies in the field, financing of research and training for producers, [17], [18]. Associated to this, the improvement of technologies in the area of communication, as well as the development of new technologies specific to IoT, made possible the reduction of the size of the hardware, optimization of energy consumption and cost reduction of devices [19].

Several reviews have been published on IoT solutions for smart agriculture in recent years which denotes that this research field is being constantly receiving new contributions and constant improvement. Existing reviews usually focus on topics like network technologies, embedded system platforms, unmanned aerial vehicles (UAV) devices, network protocols and topologies and enabling cloud platforms. For instance, [20] focuses on arable farming from year 2008 to 2018 and surveys communication technologies and protocols, the generation and analysis of data, IoT architectures and applications and highlights the challenges and future directions related with the application of IoT technologies on arable farming. Review [21] presents technologies used for communication and data collection within IoT solutions for smart farming as well as several cloud based IoT platforms used for IoT solutions for smart farming. Additionally, authors present several use cases for the identified applications of IoT for smart farming. Review [22] presents a systematic review of papers published between 2006 and 2016 and classifies these papers in application domains, such as monitoring, controlling, logistic and prediction. Within these domains, authors also identified the data visualization strategies and the technologies used for communication and edge computing. Review [23] presents a review of papers published between 2010 and 2016. The authors rely on an IoT architecture with three layers (perception, network, application) to analyze the reviewed papers in terms of perception devices, network technologies and applications. With this, they identify embedded platforms and communication technologies used in IoT solutions as well as the application of such IoT solutions. Finally, [24] reviewed papers published between 2010 and 2015 and presents a state-of-the-art of IoT solutions for smart farming and smart agriculture. Authors relied on an IoT architecture with three layers (perception, network and application) to analyze the application of sensor and actuator devices and communication technologies within several farming domains, such as agriculture, food consumption, livestock farming, among others.

1.1.Objectives and Contributions

The main objective of this dissertation is to specify an architecture of an IoT solution for smart farming capable of monitoring and acting in the mitigation of problems in plantations by collecting and processing data from crops in real-time. The novelty of this architecture is that this architecture supports both different type and extensions of plantations and enable the usage of machine learning and big data for processing the data collected by sensor nodes. This work also aims to identify how IoT is used with smart farming by (i) presenting a systematic review of the state of the art of the IoT adoption in smart agriculture and, (ii) identifying the most commonly used technologies that enable IoT solutions for smart farming.

The contributions of this dissertation comprise the proposal of an architecture of an IoT solution that allows the monitoring and correction of problems of plantations in various agricultural scenarios, such as indoor and outdoor, as well as in plantations of different sizes. The architecture proposed in this dissertation also allows the use of big data and machine learning for the processing of data collected from plantations. The architecture proposed in this dissertation is validated through the implementation of a prototype and a security risk assessment is performed to mitigate the security risks in the implemented IoT solution.

Another academic contribution of this dissertation is the publication of an article that presents a systematic review of the state of the art of IoT adoption in smart farming. The systematic review reports a change in the treatment of data in recent works: while previous work showed that the majority of decision support systems used simple processing mechanisms to handle data collected in real-time, more recent work showed an increasing number of management systems that use complementary technologies that rely on cloud and big data computing for processing large amounts of data. In terms of research domain, this work addresses the agriculture economic sector, including indoor and outdoor agriculture (greenhouse, hydroponics, crop beds, pots, orchards, permanent crops, and arable lands).

The methodology used in this dissertation consisted of the study of the state of the art of IoT solutions for smart farming to identify the most commonly used technologies and techniques that enable IoT solutions for smart farming. Subsequently, it was proposed an architecture of an IoT solution for smart farming capable of monitoring and acting in the mitigation of problems in plantations that allows the usage of the technologies and techniques identified in with study of the state of the art. The proposed architecture was then validated through the implementation of a prototype that implemented the main modules of the proposed architecture. Moreover, a security risk assessment was carried out to identify security risks related to the prototype and mechanisms for controlling and mitigation of risks were presented.

1.2.Structure of the Dissertation

To fulfill the objectives and contributions presented in the previous section, the remaining work is organized as follows:

Section 2 presents the introductory concepts necessary for understanding the work. This section starts by defining what is smart farming and presents, in brief, the technologies that can be used in this context. Subsequently, section 2 bases IoT on a 4-layer architecture and describes the characteristics of each of the layers of the architecture. Based on the 4-layer IoT solution architecture section 2 presents a study on the state of the art of IoT solutions for smart farming and identifies the main technologies and techniques that enable the use of IoT in smart farming.

Section 3 presents the proposal of architecture for an IoT solution that enables the monitoring and mitigation of problems in plantations. In this section, the general functioning of the architecture is presented. Likewise, this session also details the layers of the architecture and the components of each layer.

Section 4 describes the implementation of a prototype to validate the proposed architecture. This session first details the components used to implement the prototype. Later, the operation and communication flow of the implemented components is described. Section 4 also presents a test plan and the results of the operational tests performed. Finally, section 4 presents an assessment of security risks and the mechanisms for control and mitigation of identified security risks.

Section 5 presents the conclusions of the dissertation by showing how the objectives have been achieved and suggesting some reflections for future work.

Finally, in order to complement the dissertation, Appendix A shows the details of the security risk assessment of the prototype.

2. Theoretical Framework

This section will present the theoretical framework for the rest of the paper. The theoretical framework is divided in two parts: in the first part, the introductory concepts that are required for a better understanding of the topics discussed in this work are presented; in the second part a systematic review of IoT applied for Smart farming is presented. The results of this section were published in [25].

2.1. Introductory Concepts

This section presents the main concepts related to this work, such as smart farming and IoT.

2.1.1. Smart Farming

Smart farming is a term used to refer to several areas related to the agricultural production, such as agriculture, livestock and fishing [3], [26]. Smart farming can be understood as the use of supplementary technologies associated to agricultural production techniques in order to contribute to minimize waste and increase productivity [27], [28]. Smart farming may utilize technological resources to support in various stages of the production process, such as monitoring plantations, soil management, irrigation, pest control, delivery tracking, etc. [29]. The technological resources used in smart farming can include, for example, sensors, unmanned aerial vehicle, video cameras, agricultural information management systems, global positioning systems (GPS) and communication networks [30]. Additionally, according to [3], [26] smart farming may use sensors to collect data in real time from different rural production areas. These data allow interventions in the production process to be performed in exact time, quantity, and location [15]. Furthermore, smart farming considers other technologies (e.g., big data, business management systems, etc.) to provide a more comprehensive panorama in terms of location, context and situation of the entire production [3], [26].

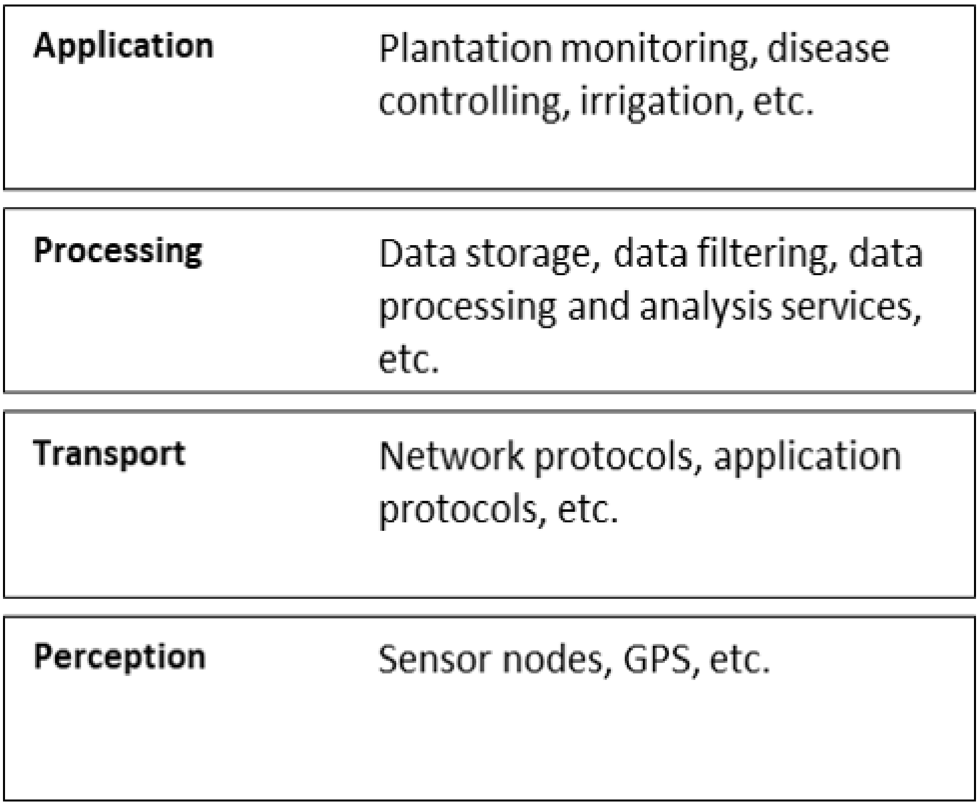
2.1.2. Internet of Things

IoT can be understood as a network of interconnected intelligent devices capable of communicating with each other, generating relevant data about the environment in which they operate. Thus, virtually any device capable of establishing a connection to the Internet can be considered a “thing” within the context of IoT, such as household appliances, electronics, furniture, agricultural or industrial machinery and even people [19].

Although the idea of IoT is not new, its adoption has increased in recent years, mainly thanks to the development of technologies that support it, among which the improvement of hardware – with the consequent reduction in size and power consumption – improvements in connectivity with the Internet and between devices via wireless connection, cloud computing, artificial intelligence and big data. All these technological components help build a network of devices capable of sharing data and information, as well as acting actively based on network inputs [31].

According to [32], the architecture of IoT systems is similar to the architecture of other computer systems but it must take into account the particularities of this paradigm, such as the limited computing capabilities of the devices, identification, detection and control of remote objects.

The IoT architecture proposed in [33], [34] and shown in Figure 2.1 presents four layers, considering the main components of an IoT solution: devices, network, services, and application.



Application	Plantation monitoring, disease controlling, irrigation, etc.
Processing	Data storage, data filtering, data processing and analysis services, etc.
Transport	Network protocols, application protocols, etc.
Perception	Sensor nodes, GPS, etc.

Figure 2.1 – 4-layers IoT solution architecture, based on [33], [34].

The perception layer relates to the physical devices in the solution and how they interact with each other and with the transport layer. These devices are responsible for collecting data, enabling the communication of the so-called “things”. This can be done by using commercial solutions – such as UAV devices [35], sensor nodes [36]– or new devices, developed with components like sensors and single-board computers (SBC) – such as Arduino or Raspberry Pi – to build sensor nodes and communication gateways. Sensor nodes, for example, are used to monitor plant diseases [37], control environmental variables in greenhouses [38] and external crops [39]–[41], among others. The interaction between the devices that belong to the perception layer and the services that belongs to the processing layer is intermediated by the transport layer and might occur in several ways, such as through the direct communication between sensor nodes and a data processing platform (such as FIWARE [42], SmartFarmNet [43] and Thinger.io [44]) or through a gateway that, besides intermediating the communication between sensor nodes and the internet, acts as a data hub and enables the communication between network protocols that are originally incompatible, such as ZigBee and the Internet [39].

The transport layer refers to the network and transport capabilities such as network and application protocols [33]. IoT solutions use network protocols to enable communication between the perception layer and the processing layer. These protocols are used to create the so-called wireless sensor networks (WSN), that allows wireless communication between sensor nodes and applications. Each protocol has important characteristics, such as the data exchange rate, range, and power consumption. Based on these characteristics such protocols can be classified in short-range, cellular networks and long-range [45]. Protocols for short-range networks (e.g., Bluetooth, ZigBee, and Wi-Fi) enable communication in short distances. According to [45], usually such protocols have a high data transmission rate and low power consumption. Therefore, they are used for the communication between devices that are near each other. Protocols for cellular networks (e.g., GPRS, 3G) enable communication in long distances and with a high data transmission rate. However, they have a high power consumption [46] and costs for licensing [45]. Protocols for long-range networks (e.g., LoRaWAN and Sigfox) enable communication in very long distances [45]. These protocols are used to establish the low power wide area networks (LPWAN) due to the fact that they have a low power consumption [47]. However, the data transmission rate of these protocols is low. Therefore, these protocols are appropriate for use when the solution needs to transmit a few amounts of data in very long distances. Table 2.1 presents the characteristics of some network technologies used for IoT.

Table 2.1 – Examples of network technologies used in IoT [45].

Parameter	Wi-Fi	Bluetooth	ZigBee	LoRa
Standard	802.11 a, b, g, n	802.15.1	802.15.4	802.15.4 g
Frequency	2.4 GHz	2.4 GHz	868/915 MHz, 2.4 GHz	133/868/915 MHz
Data rate	2–54 Mbps	1–24 Mbps	20–250 kbps	0.3–50 kbps
Transmission Range	20–100 m	8–10 m	10–20 m	>500 m
Topology	Star	Star	Tree, star, mesh	Star
Power Consumption	High	Medium	Low	Very Low
Cost	Low	Low	Low	Low

As shown in Table 3.1 there is a trade-off between coverage, data rate and energy consumption. Considering the technologies for star networks presented in Table 3.1, it is possible to notice that energy consumption is higher in technologies with a high data rate and short coverage. On the other hand, LoRa has a small data rate but a large coverage and low power consumption.

These questions are especially relevant when considering agriculture because agricultural scenarios often have limited or no energy supply and obstacles for wireless communication.

Different topologies can be used for implementing networks, such as tree, star, and mesh. Star networks have a central node and several peripheral nodes. The communication in such topology occurs as follows: peripheral nodes send data directly to the central node. The central node can implement capabilities for routing messages and communicating through multiple network protocols [34]. Tree networks are composed of router nodes and leaf nodes. Such networks can be understood as a cluster of star networks. Within each cluster, leaf nodes send messages their father node. In mesh networks, in theory, each node can be a router with rerouting capability. Thus, messages in mesh networks are routed hop by hop until reaching the final destination [48].

Data are sent to the destination through application protocols such as the message queueing telemetry transport (MQTT) [49] or the constrained application protocol (CoAP) [50]. MQTT is an open-source messaging protocol that enables communication between constrained devices and in unreliable networks [51]. The MQTT protocol runs over TPC/IP or similar protocols (e.g., Bluetooth) [52], which makes the use of MQTT protocol appropriate for different IoT solutions. The MQTT protocol, which is based on the publish/subscribe architecture, allows communication between devices to take place in the following way. First, devices publish messages that are structured in topics on a message broker. Then, other devices read these messages by subscribing to relevant topics on the message broker. These topics allow the organization of messages based on categories, subjects, etc. [53]. The use of MQTT protocol for communication between device allows low coupling between the device that publishes the message and the devices that listen to the messages, the so-called "one-to-many" communication [49]. Like MQTT, CoAP is a communication protocol optimized for constrained devices and unreliable networks. However, CoAP messages are interchanged using User Datagram Protocol (UDP) and the CoAP protocol is based on the client/server architecture. This architecture requires that a connection is established between devices before any messages are transmitted [49]. For this reason, communication using CoAP works in the following way. First, the device that sends messages needs to know the address of each device that is expected to receive messages. Then, messages are sent over UDP to the specified address. Due to the use of UDP, CoAP messages are classified accordingly to the required status of confirmation of receival, for example, confirmable or non-confirmable [50]. The CoAP protocol does not implement a structure of topics for messages. However, a similar approach can be implemented using application programming interface (API). Nonetheless, the use of CoAP creates a high coupling between the device that sends messages and the device that is expected to receive messages, as the communication is “one-to-one” [50].

The processing layer comprises data storage, visualization, and processing resources. In this context, big data allows distributed storage and parallel data processing, enabling the extraction of information in the shortest possible time [54]. Such information are used as models by artificial intelligence (AI) systems – which, according to [55], can be understood as the ability of a system to operate as if it had the thinking capacity of a human being – and machine learning – that, according to [56] is a data processing technique to detect patterns and correlation among complex and unrelated data – for the development of decision support systems and automation of irrigation control systems [57], monitoring [58] and diseases detection in crops [59], for example.

Finally, the application layer comprises IoT applications that, supported by the other mentioned layers, provide management information to farmers, being able to manage the entire production process in the plantations.

2.2.A Systematic Review of IoT Solutions for Smart Farming

As presented in section 1, several related works are being developed in recent years. This rich literature has already been analyzed by the academia from multiple perspectives with objective of determining the state of the smart farming development. Thus [60] presented a systematic review of precision livestock farming in the poultry sector and [61] made a review of state of the art of technologies used in precision agriculture, focusing in the innovations, measured parameters, technologies and application areas. On the other hand [3] has focused on the use of big data as a tool to support agriculture, pointing out the main opportunities and challenges of using this technology. Finally, [62] presented a quantitative literature review on smart farming related papers, helping to outline an overview of academic production related to the subject. In this way, the present work aims to complement such analyses by making a systematic review of IoT solutions applied to smart farming.

2.2.1. Methods for the Systematic Review

To reach the proposed objectives, this study has used the Preferred Reporting Items for Systematic Reviews (PRISMA) methodology, which is a framework developed to support reports and systematic reviews of literature [63].

As a research strategy, in October 2019 a search was made in the Scopus database through the search tool available on the website. In addition, in June 2020 a new search was made in the same database to include papers published in 2020. The choice of this database took into consideration its scope and relevance in the academia, since this database indexes several journals and catalogues, such as IEEE, ACM and Elsevier, besides being widely used in similar bibliographic reviews, as in [3] and [62]. In addition, in February 2020 a new search was performed in the same database. The strategy adopted for the work research in this database looked for terms used to refer to the application of technology in the area of agriculture, such as “Precision Agriculture”, “Precision Farming”, “Smart Farming” and “Smart Agriculture” in association with “IoT” and synonyms terms. The publication date of the articles was not a criterion for ignoring them. The scope of the research was limited to documents such as journal and conference articles, published in English, Portuguese or Spanish, and whose access was fully available. Thus, the resulting search instruction for the database was as follows:

```
("Smart Farming" OR "Smart Agriculture" OR "Precision Farming" OR "Precision Agriculture") AND ("IoT" OR "Internet of Things" OR "internet-of-things") AND (LIMIT-TO(ACCESSTYPE(OA)))
```

It should be noted that the quotation marks have the function of ensuring that terms composed of multiple words were searched together, thus preventing words from being considered individually.

After extracting the articles that resulted from the search, they were manually reviewed through the analysis of the title, keywords, abstract and text. Initially, based on this review, the works identified in the researched database were consolidated, thus eliminating duplicate articles.

Subsequently, the articles were validated as to their framing in the objectives proposed for this study and considered valid when: (i) they were not a review or bibliographical research (ii) they were related to theme (iii) they presented a technology or solution based on IoT to solve problems related to agriculture (iv) they were published in English, Portuguese or Spanish. Furthermore, works were also excluded when they were related to livestock activities instead of agriculture.

The process of searching and selecting papers for this study followed the workflow summarized in Figure 2.2, where it can be observed that the initial search resulted in a total of 463 articles, which were analyzed, filtered and classified in a narrowing process that culminated in the selection of 159 articles.

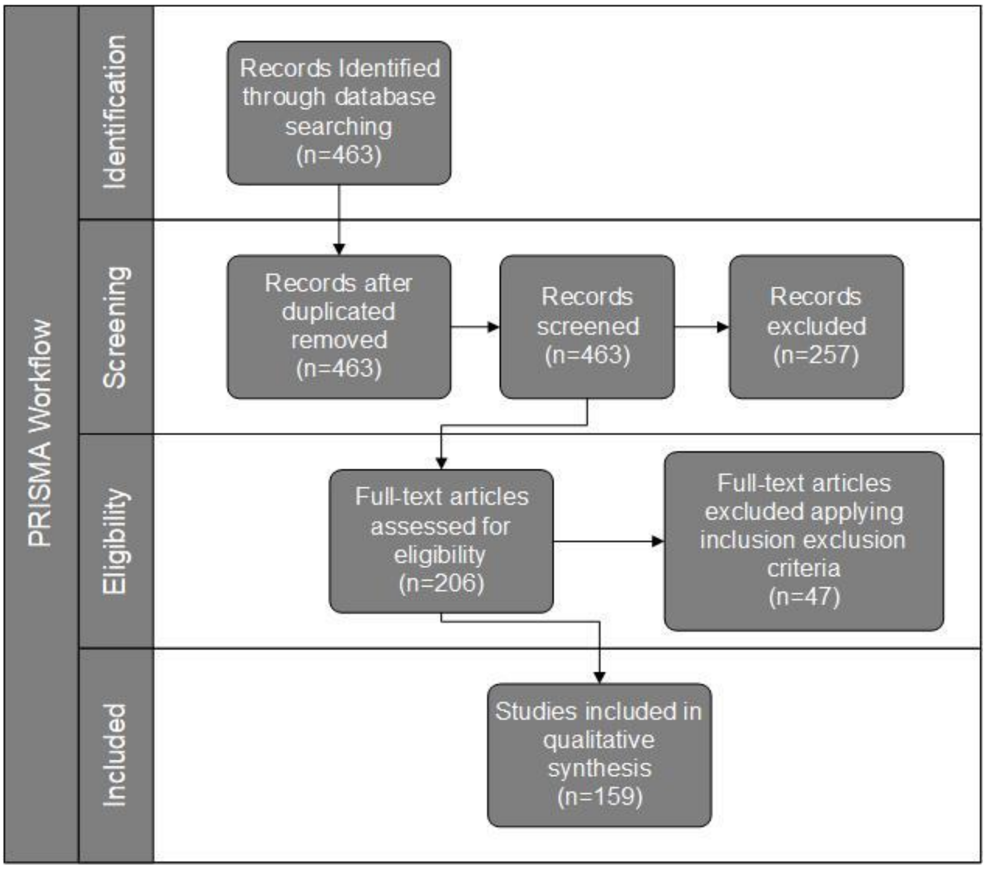

PRISMA Workflow
Identification
Screening
Eligibility
Included
Records Identified through database searching (n=463)
Records after duplicated removed (n=463)
Records screened (n=463)
Records excluded (n=257)
Full-text articles assessed for eligibility (n=206)
Full-text articles excluded applying inclusion exclusion criteria (n=47)
Studies included in qualitative synthesis (n=159)

Figure 2.2 – PRISMA flowchart of the systematic review on state-of-the-art IoT solutions.

In the identification phase 463 articles were selected with the search tool.

During the screening phase, a manual review of the articles was carried out to identify in the titles, abstract and key words the papers adherent to the objectives proposed for this study, following the criteria mentioned in this section. Among these, 257 were considered invalid and discarded. About 62% of the discarded items did not consider smart farming to be the focus of the work, although some presented improvements for IoT that could benefit smart farming indirectly. Additionally, almost 31% of the discarded papers were studies or literature reviews related to smart farming and the use of various technologies. A smaller number of papers related to smart farming but not addressing IoT (about 5%) and papers where the abstract or text were not available (about 2%) were also discarded.

During the eligibility phase, the content of the 206 resulting articles were reviewed and the papers were classified using the same criteria used in the previous step. In this phase 47 articles were discarded. Among the discarded articles 29% were not related to IoT and 30% were not related to smart farming. The other 41% of the discarded papers were paper reviews or papers without content available. This analysis resulted in 159 articles considered eligible which were included as a sample for this study.

2.2.2. Discussion

Based on the results obtained in the analysis of the articles considered for this study, it was possible to observe a growth trend in the number of publications related to IoT and smart farming since 2011, with special emphasis from 2016 onwards, as shown in Figure 2.3.

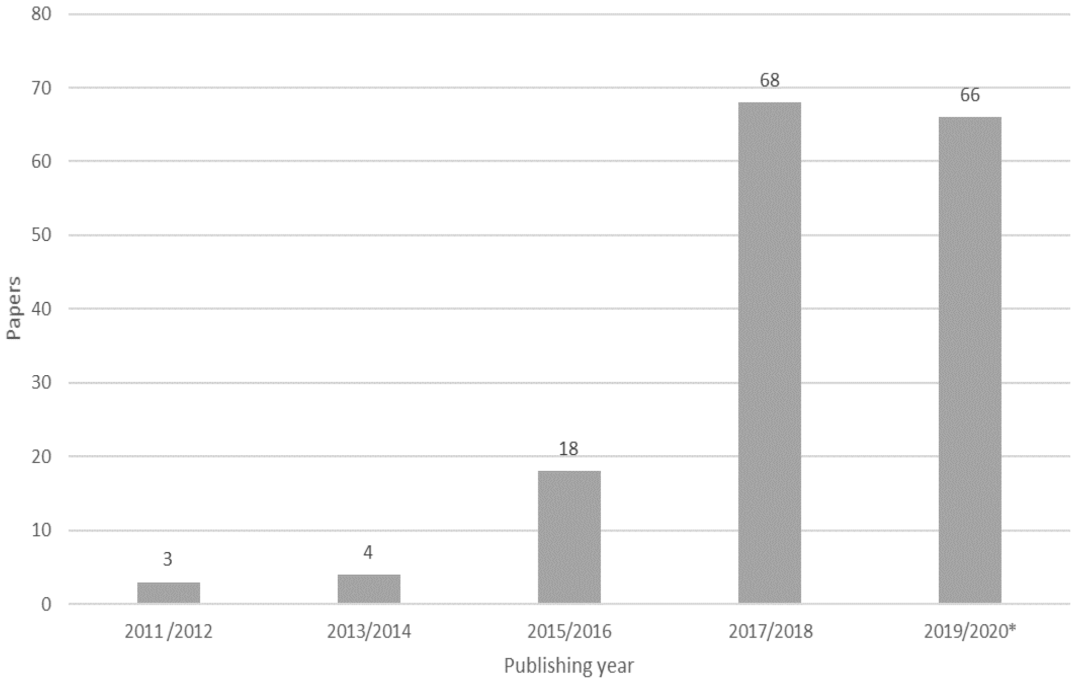


Figure 2.3 – Classification of reviewed papers according to the year of publication.

It is possible to observe an expressive increase of 278% in the number of published papers in 2017/2018. It is also possible to observe a very similar number of published papers in

2019/2020, until the first semester of 2020. The amount of published papers in recent years evidences the increasing in discussion and the relevancy of the topic IoT applied to smart farming.

Within the reviewed papers it was identified the main scenarios and environments of agriculture. As shown in Figure 2.4, such scenarios can be divided into indoor and outdoor. Environments for indoor scenario are protected from climatic impacts, such as solar radiation, rain, and wind. Examples of environments for indoor scenarios include greenhouse, hydroponics, crop beds, pots, etc. In contrast, environments for outdoor scenario are more susceptible to climatic impacts. Examples of environments for outdoor scenario are arable lands, orchards, and generic outdoor plantation**.**

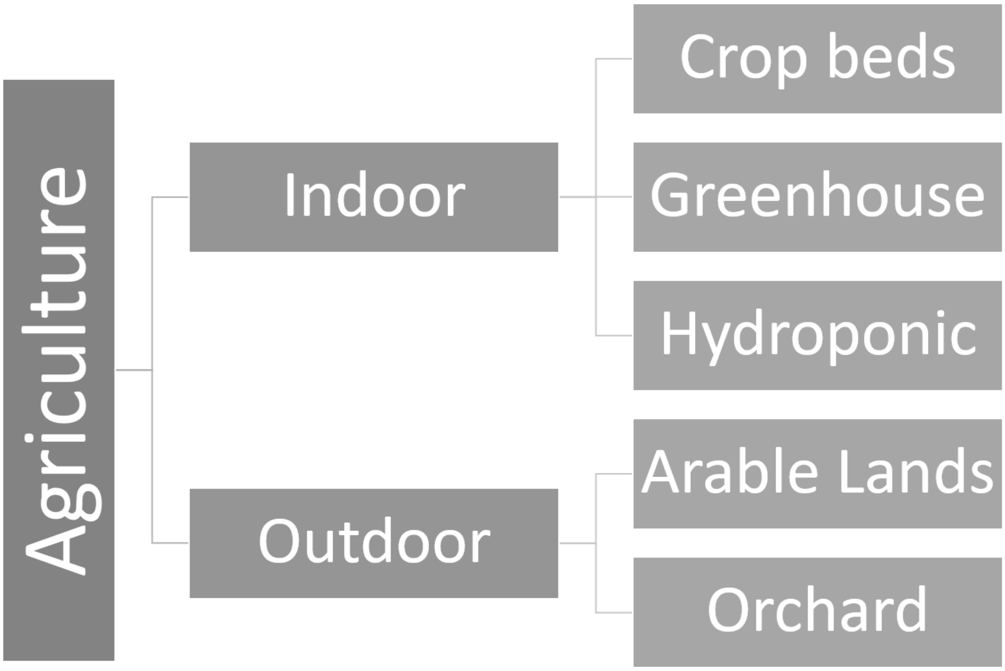


Figure 2.4 – Typical agricultural scenarios and environments.

2.2.2.1. Application

Within the reviewed papers it was also identified that the most common applications of IoT solutions for smart farming are:

- Chemical control (e.g., pesticides and fertilizers).
- Crop monitoring.
- Disease prevention.

- Irrigation control.
- Soil management.
- Supply chain traceability.
- Vehicles and machinery control.

Table 2.2 presents the reviewed papers, grouped by agricultural environment and application of the IoT solution. It is worth mentioning that several IoT solutions presented on the reviewed papers could be applied to multiple environments (Figure 2.3). Thus, such IoT solutions are classified as “Generic”. Additionally, the “Others” column in Table 2.2 includes papers whose IoT solutions were developed for agricultural environments that were less mentioned, such as pots, crop beds, etc. It is possible noting a predominance in projects where the application is for crop monitoring, irrigation management, and disease prevention.

Table 2.2 – Smart farming, applications, and environments.

Application	Arable Land	Generic	Greenhouse	Orchard	Other
Chemical control	[10], [64]	[59]	[65]	[66]	
Crop monitoring	[12], [13], [70]–[79], [35], [80]–[86], [36], [41], [49], [58], [67]–[69]	[40], [44], [95]–[104], [87], [105], [106], [88]–[94]	[107], [108], [117]–[126], [109], [127]–[130], [110]–[116]	[14], [131]–[133]	[134], [135]
Disease Prevention	[136]–[139]	[82], [140]–[143]	[11]		[9]
Irrigation control	[39], [144]–[149]	[38], [57], [150]–[153]	[154], [155]	[45], [156]	[157]–[160]
Soil Management	[161]	[162]–[165]		[166], [167]	[168]
Supply chain traceability		[169], [170]	[171]		[6], [172]–[175]
Vehicles and machinery control					[5], [52], [176], [177]
Other	[178], [179]	[180]–[182]	[183]	[184]	[43], [185], [186]

As shown in Table 2.2, the most common application of IoT solutions for smart farming is crop monitoring. Moreover, as shown in Table 2.2, these solutions have been developed for multiple agricultural environments, such as arable lands, orchards, greenhouses, etc. The fact that this type of application is so common in agriculture can be justified by the relevance that crop monitoring has for farmers. IoT solutions developed for monitoring crops focused on collecting environmental data of plantations (such as temperature, humidity, luminosity, etc.). Farmers can use these data to obtain a better insight of the plantations. For example, such data was used to determine the vigor of rice [13], [58], alfalfa [41] and maize [67] crops and to control the environmental conditions of greenhouses [107], [108], [110], [112]. Similarly, IoT solutions for irrigation control has also been developed for multiple agricultural environments, as demonstrated in Table 2.2. Such IoT solutions aimed to optimize the use of water resources in agriculture in different ways, such as by simply using sensors for measuring the soil moisture and using these data for controlling the irrigation source [45], [144] or in a more sophisticated way, by combining humidity data with datasets of weather to determine the amount of water required during the irrigation [145]. IoT solutions for disease prevention aimed to identify and prevent diseases on plantations. For this purpose, these IoT solutions collected multiple environmental and plantation data, such as images of plants [136], [139], [141], sounds [142], temperature, humidity, etc. [11], [138]. These data were processed with different approaches, such as image processing [136], [141] or artificial intelligence [11], [139]. For example, the IoT solution developed in [136] processes images collected from a sugarcane crop and identifies diseases on the leaves of plants. In addition, [142] developed an IoT-enabled device that captures sounds produced by larvae inside trees. IoT solutions for chemical control presented in Table 2.2 aimed to optimize the application of fertilizers and pesticides on plantations. For this purpose, these IoT solutions collect data (such as nitrogen, salinity, or pH) from the crops. Based on the collected data, such IoT solutions can identify crop areas that may require the application of fertilizers or pesticides. For example, in [64] aerial images of crops are processed to determine the nitrogen concentration in a large plantation. These images are useful to determine the specific region that requires fertilizer. In addition, [65] developed an automated robot that optimizes the application of pesticides in greenhouse cultivations. IoT solutions for soil management aimed to identify different soil attributes used for planting. For example, such IoT solutions are used to measure the soil moisture [168], to identify the water consumption pattern [164], [166] and to identify the nutrients of the soil [163]. IoT solutions for vehicles and machinery control focused on collecting data of and managing agricultural equipment and machinery such as tractors, harvesters, and trucks. For this purpose, IoT solutions had to deal with the characteristics inherent to agricultural equipment, such as mobility. Data from the equipment itself, such as implement status, engine performance, or speed are collected using sensors [52] to optimize their maintenance cycle. Additionally, due to the mobility of agricultural equipment, opportunistic computing was used to collect data from remote crop areas by using sensors coupled to tractors [5].

Each agricultural environment presented in Table 2.2 brings its own challenges for the projects, which includes the environment impact on the communication between sensors, either by the distance between the sensor nodes [36], [113], [183], by the lack of communication in the croplands [5], [106] or even by the impact of vegetation in the signal propagation [78], [178]. Furthermore, as indicated in [39], climatic elements – such as rain, snow or solar radiation – have influence on both the planting and the sensor nodes.

To cover these scenarios commercial electronic sensors are used by 96% of the reviewed papers. This expressive usage can be justified by the fact that such sensors are affordable, certified, ready-to-market and meet the main monitoring needs in IoT solutions for smart farming. Such sensors are used for collecting real-time data about multiple agricultural parameters, such as climatic data, substrate information, luminosity, CO_2 concentration and images through cameras and multispectral sensors, as shown in Table 2.3. Moreover, several papers (4%) focused on developing custom-made sensors for monitoring specific agricultural aspects, such as soil nutrients (e.g., nitrate [163]) and leaf evapotranspiration for measuring the hydric stress in tobacco crops [89].

Table 2.3 – Types of physical sensors and use in smart farming.

Use in Agriculture	Application of Sensors	Examples of Sensors (Models)	References
Crop monitoring	Growth	Cyber-shot DSC-QX100 (Sony Electronics Inc., Tokyo, Japan), Parrot Sequoia (MicaSense Inc., Seattle, WA, United States)	[14], [67], [133]
	Insects and disease detection	FLIR Blackfly 23S6C (FLIR Systems, Wilsonville, OR, USA)	[82], [141], [187]
	Active canopy sensor	ACS-430, ACS-470 (Holland Scientific, Inc., Lincoln, NE, USA)	[13], [58], [64]
Substrate monitoring	Soil temperature, soil moisture	DS18B20 (Maxim Integrated, San Jose, CA, USA), VH400 (Vegetronix, Salt Lake City, UT, USA), HL-69, ECH2O-10HS (METER Group, Pullman, WA, USA)	[66], [69], [88], [101], [156]
	pH	E-201 (Shanghai REX Sensor Technology Co, Shanghai, China)	[104]
	Chemical elements (e.g., nitrate, nitrogen, etc.)	SEN0244 (DFROBOTS, Shanghai, China)	[117]
Environment monitoring	Air temperature, air humidity	DHT11, DHT22 (AM2302, Aosong Electronics Co. Ltd., Guangzhou, China)	[112], [122], [159]
	Solar radiation	SQ-110 (Apogee Instruments, Inc., Logan, UT, USA)	[113]
	Rain	YF-S402 (Graylogix, Bangalore, Karnataka, India), YL-83 (Vaisala Corp., Helsinki, Finland) SE-WS700D (Lufft Inc., Berlin, Germany)	[38], [132]

	Luminosity	BH1750 (Rohm Semiconductor, Kyoto, Japan), TSL2561 (Adafruit Industries, New York City, NY, USA)	[9], [117]
	Atmospheric pressure	MPL3115A2 (NXP Semiconductors, Eindhoven, Netherlands)	[49]
	Wind speed and direction	WS-3000 (Ambient Weather, Chandler, AZ, USA), SEN08942 (SparkFun Electronics, Niwot, Colorado, USA)	[49], [113]
	CO2 concentration	MG-811 (Zhengzhou Winsen Electronics Technology Co., Ltd., Zhengzhou, China), MQ135 (Waveshare Electronics, Shenzhen, China)	[104], [112]
Other	Tracking	Mifare Ultralight NFC tag (NXP Semiconductors, Eindhoven, Netherlands), Blueberry RFID reader (Tertium Technology, Bangalore, Karnataka, India)	[6], [173]
	Localization	UM220-III (Unicore Communication Inc., Beijing, China)	[83], [184]

As presented in Table 2.3, different types of sensors were used in IoT solutions for smart agriculture to collect data from multiple aspects of agriculture, such as the crop, substrate, environment and other. For this purpose, as shown in Table 2.3, for environment monitoring electronic sensors were used in IoT solutions to collect environmental data, such as temperature, humidity and luminosity [112], [117], [122]. In addition, for substrate monitoring electronic sensors were used to collect data from the substrate (e.g., soil and water), such as temperature, moisture, and nitrogen. Likewise, pH sensors were used for measuring the acidity or the alkalinity of the water in hydroponics cultivations. For crop monitoring, cameras and multispectral sensors were used to collect images of crops. These sensors can be installed on an UAV to obtain aerial images of large plantations [13], [58], [67] or used in robots to retrieve a detailed image of the leaf of a plant [119].

2.2.2.2. Perception

The choice of hardware is an important aspect of the IoT project development because it impacts the costs and the technologies that can be used. 60% of the reviewed papers mentioned the hardware used to support the IoT solution. Furthermore, SBCs were mentioned by 40% of the reviewed papers. The use of SBCs can be justified by the fact that these devices are affordable and versatile [49], enabling the development of custom-made IoT devices. For example, some SBCs such as Arduino has an integrated development environment (IDE). This

IDE enables the development of custom programs to be installed as firmware on the Arduino boards [188]. Similarly, Raspberry Pi is compatible with several operating systems, such as Raspbian, Ubuntu Core or Mozilla Web Things [189]. Some of these operating systems are open-source, which allow for the customization of its source-code. Besides, these operating systems support applications developed with programming languages such as Python [37]. Furthermore, the capabilities of SBCs can be extended by associating them with other hardware components, such as sensors or transceivers. This characteristic makes SBCs able to work as gateways or sensor nodes in IoT solutions. Among the papers that mentioned SBCs, 82% mentioned the use of Arduino, Raspberry Pi and ESP boards (such as ESP8266, ESP12 and ESP32). Table 2.4 presents the application of embedded system platforms and UAV devices in smart farming.

Table 2.4 – Embedded system platforms and UAV devices in smart farming.

Application	Arduino	Raspberry	ESP	UAV
Disease prevention	[136]–[139]	[137]	[136], [137]	[136]
Waste management		[150]		
Chemical control				[10], [59]
Crop monitoring	[12], [35], [115]–[118], [123], [124], [126], [128], [130], [135], [71], [73], [92], [102], [106], [109], [113], [114]	[40], [41], [125], [126], [92], [103], [106], [118]–[122]	[44], [71], [88], [105], [110], [112], [124], [127], [135]	[13], [14], [35], [58], [67], [74], [84], [190]
Soil management	[164], [167]	[41], [161]		
Vehicles and Machinery control		[5]		
Irrigation control	[38], [45], [144], [152], [159], [160]	[38], [149]	[151], [159], [191]	

As shown in Table 2.4, IoT-enabling devices are used for multiple applications on IoT solutions for smart farming. SBCs were used both as sensor nodes and gateways. Table 2.4 reveals that Arduino was the most commonly used embedded system platform among the reviewed papers. The extensive use of Arduino can be justified by the fact that Arduino is open-source hardware that enables the development of different devices through the use of boards that extend their native functionality. Table 2.4 also shows that embedded system platforms have been more widely used in IoT solutions for crop monitoring. As sensor nodes, for example, in [132] sensors for collecting environmental data such as soil humidity, solar radiation and rain are connected to an Arduino Uno. The Arduino is, then, used to monitor the health of a vineyard. Likewise, in [125] a Raspberry Pi is used to manage the temperature and air humidity of a greenhouse. IoT devices are also used as gateways to connect short-range WSN with the internet by using long-range communication protocols. For example, in [134] a gateway is used to connect WSNs using 3 different protocols (ZigBee, Bluetooth and Wi-Fi) with a remote server by using 3G. In [92] a LoRaWAN gateway obtains data from sensor nodes using LoRa and retransmits this data to a cloud-hosted platform by using 4G. 3G and 4G are cellular network technologies that, as discussed in section 2.1.2, enable communication in long distances and with a high data transmission rate. These technologies will be discussed with more details in section 2.2.2.3.

In addition, Table 2.4 also reveals that UAV is widely used by IoT solutions for monitoring crops, disease prevention and chemical control. The use of UAV for crop monitoring is due to the fact that UAV has the potential to accelerate and reduce the cost of monitoring extensive crops. For this purpose, cameras and multispectral sensors are attached to UAV devices that are used to obtain aerial images from large crops. Such images are processed by the IoT solution to calculate agricultural parameters, such as the leaf area index (LAI). The LAI is a parameter used to determine the vegetation coverage within a specific area. LAI, combined with other parameters, can be used to evaluate the amount of nitrogen in rice crops [13], determine the vigor of rice and maize [58], [67] crops and detect diseases in sugarcane crops [136]. Moreover, UAV devices are used in [57] to optimize the application of pesticides and fertilizers in arable lands.

2.2.2.3. Network

Data obtained with sensor nodes are usually sent to the destination (e.g., database, server, IoT platform) through a wired or wireless network. Within the reviewed papers, 60% have mentioned the network protocol used in the IoT solution. Among the mentioned network protocols, CAN and Ethernet were the most used ones for wired networks. Likewise, LoRaWAN and protocols for cellular network (e.g., GPRS, 3G, etc.) were the most used protocols for long-range wireless networks. Analogously, ZigBee, Wi-Fi and Bluetooth were the most used protocols for short and mid-range wireless networks. Table 2.5 shows network protocols used for the IoT solutions within the reviewed articles.

Table 2.5 – Use of network protocols in smart farming for different farming scenarios.

Network Protocols		Arable Land	Generic	Greenhouse	Orchard
Wired	CAN		[150]	[108]	[66]
	Ethernet		[82], [92]	[118], [130]	

Short range	Bluetooth	[74]	[89], [91], [191]	[116], [123], [125]	[192]
	LoRa	[69], [75], [76], [146]	[106], [191]	[11], [122]	[45]
	NFC				[192]
	RFID		[104], [162], [170], [182]		
	ZigBee	[12], [39], [41], [78], [138], [178]	[91], [100], [101], [104], [163], [181], [182], [193]	[107], [113]–[115], [120], [121], [126]	[66], [167], [192]
Middle Range	(RF-ISM)	[35], [77], [78], [149]	[152], [182], [193]	[111], [171], [183]	[132]
	Wi-Fi	[71], [136], [137], [139], [144], [161]	[38], [44], [180], [82], [88], [91], [92], [102], [104], [105], [151]	[109], [110], [124], [125], [127], [128], [112], [113], [115]–[117], [119], [121], [123]	
Long range	LoRaWAN	[76]	[92], [98], [106], [142]	[11], [122]	
	Cellular	[39], [49], [74], [83], [136], [144], [146], [147], [179]	[9], [106], [140], [164]	[107]–[109], [117], [125]	[66], [132], [156]
	Sigfox				[45]

As shown in Table 2.5, several network protocols are used in different environments of agriculture (e.g., arable land, greenhouse, orchard) to enable communication between IoT solution devices, such as sensor nodes and gateways. Such network protocols enable the creation of short or long-range networks. Table 2.5 reveals that for short and middle-range communication, IoT solutions of the reviewed papers used different technologies, such as Wi-Fi, ZigBee, and Bluetooth. Moreover, it is possible to observe in Table 2.5 that Wi-Fi is the most common network technology for communication within the analyzed articles. This extensive use of Wi-Fi can be justified by the fact that Wi-Fi is a ubiquitous technology and, therefore, easy to implement. However, due to the higher energy consumption of Wi-Fi, low-energy consumption technologies, such as ZigBee or Bluetooth, are also extensively used. For example, [12] used ZigBee to send images from a plantation to a remote server and [192] developed a sensor node that uses Bluetooth to deliver monitoring information from the farm directly to an application installed on a smartphone. Table 2.5 also demonstrates that IoT solutions of the reviewed papers used cellular networks, Sigfox, or LoRaWAN for long-range networks. Cellular networks are prevalent in IoT solutions for Smart Farming. This can be justified by the fact that cellular networks allow the communication of IoT devices in long distances and with a high data rate. For example, [146] uses cellular network to send data collected from humidity sensors to a cloud-based platform and to control an irrigation system.

Similarly, Sigfox and LoRaWAN enable communication in very long distances while requiring low energy to operate. Based on these characteristics, Sigfox and LoRaWAN were used for long-range communication, as an alternative to cellular networks or in regions where there was no cellular network coverage. Sigfox is used in [45] as the network protocol of an IoT solution used to control the irrigation of a plantation. Likewise, in [11] the LoRaWAN is used to send data from multiple sensors installed in a greenhouse to a remote platform.

Besides the distance between sensor nodes, gateways, and other network elements, the vegetation itself can be an obstacle for sensor communication, as demonstrated by [178] and [78] who analyzed the impacts on signal propagation on 433 MHz and 2.4 GHz frequencies in rice plantations and an orchard. An additional challenge for greenhouses arises from the high density of sensors, which can lead to interference in the wireless signal due to proximity [113], [120], [183]. To mitigate this problem wired networks, such as CAN [108] or Ethernet [130], can be used. As shown in Table 2.5, these technologies have been more used in greenhouses, because usually this type of agricultural environment is more appropriated for implementing wired networks. Moreover, [120] investigated the path loss on wireless signals and concluded that the proper positioning of directional antennas can optimize the number of sensory nodes required for monitoring a greenhouse.

Network topology is another important aspect of an IoT solution. According to [70] the topology of sensor networks can be star, tree (or cluster) or mesh. The network topology impacts the distance between the sensor nodes and the destination and, consequently, the number of sensor nodes in the WSN [194]. For example, star networks are composed of a central node (coordinator) and several peripheral nodes. In such topology, peripheral nodes send data to the central node [101]. Therefore, the maximum distance between the peripheral nodes and the central node is limited by the maximum distance allowed by the physical layer communication standard. On the other hand, as discussed in section 2.1.2, in mesh networks each node has routing capability, hence extending the network coverage by allowing multi-hop communications [195]. Based on the architecture of the IoT solution and on the project description it was possible to identify the topology adopted by 61% of the reviewed papers. For example, a star topology is used in [45] for connecting sensor nodes to a central node using the LoRa protocol. This central node acts as a gateway and retransmits messages to a cloud-based application that controls an irrigation system using Sigfox. Also, in [115] the star topology is used to connect multiple sensors within a greenhouse. Such sensors use the ZigBee protocol to send messages to a central node, which acts as the network gateway. Mesh networks are considered more complex to be implemented but also more reliable due to the redundancy of communication between the sensor nodes [113]. Such topology is used in [113], [115] for monitoring a greenhouse. Tree (or cluster) networks combine multiple star networks. Both [70] and [181] implement a cluster network for monitoring crops. In [70] sensor nodes collect information from a crop and send messages to a router node. This router node acts as the gateway of the cluster and retransmits the message to the main router node of the network. In [181] several router nodes are deployed in the crop area in order to optimize the energy consumption of sensor nodes.

Furthermore, embedded system platforms have been used to support network topologies. The chart in Figure 2.5 presents the distribution of embedded system platforms by network topology or device connection type. It is worth mentioning that although point to point is not a network topology, this type of device connection was used in several IoT solutions within the review articles. As shown in Figure 2.5, Raspberry Pi is often used in IoT solutions implementing the star network topology. Arduino is the embedded system platform used in multiple types of network topology or device connections. Additionally, Arduino is the most frequently used embedded system platform to support star network topology and point-to-point communication. Finally, ESP-based devices include devices that use system-on-a-chip (SoC) modules such as ESP-32 and ESP8266 (Espressif Systems, Shanghai, China). ESP-based devices are often used in IoT solutions that implement star network topology or point-to-point communication.

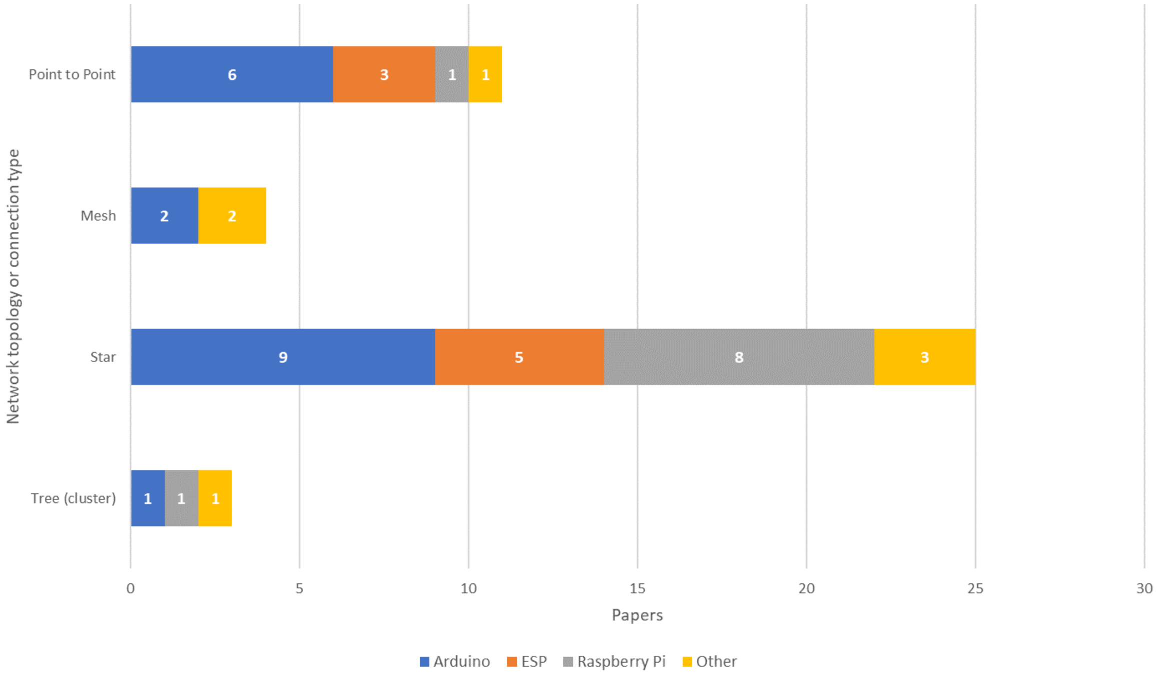


Figure 2.5 – Distribution of IoT-enabling devices by network topology or device connection type within the reviewed papers.

As mentioned in section 2.2.2.2, embedded system platforms can be used to build gateways or sensor nodes. As shown in Figure 2.5 the use of Raspberry Pi, Arduino and ESP stand out, probably because such embedded system platforms are cost-effective [49] and enable different network protocols (e.g., ZigBee, Wi-Fi and Bluetooth) with the use of transceivers. This characteristic allows such embedded system platforms to act as sub-nodes and central nodes in a star network [38], [41], [45] or as router nodes in mesh and cluster networks [114], [181].

IoT devices transmit information to cloud-based platforms or applications through application protocols [117]. Such protocols can follow the publisher/subscriber architecture which, as mentioned in section 2.1, are appropriate for devices with limited computing resources. Among the application protocols used in the reviewed papers HTTP, MQTT and CoAP stand out. Such application protocols are useful to enable compatibility between non-standardized IoT devices and IoT platforms. For example, SmarFarmNet developed in [43] adopts the “bring your own IoT device” concept by implementing loosely coupled application protocols such as MQTT and CoAP. Furthermore, although HTTP is not a specific protocol for machine-to-machine (M2M) communication, its use associated with REST APIs enables low coupling between IoT devices and applications, analogous to MQTT protocol, for example. However, as [117] concludes, the

MQTT protocol is preferable for smart farming applications due to its resiliency, interoperability across different network protocols and transmission rate.

Finally, although the power consumption is not an exclusive topic within the transport layer, according to [181] the highest power consumption for IoT devices within a WSN occur during the transmission of data. This review identified several approaches for optimizing the power consumption in IoT solutions for smart farming. Among the identified solutions are the use of low energy protocols (e.g., BLE, ZigBee, Sigfox), reduction of data transmission in sensor nodes by an optimized duty cycle [180], [181], [196] and the use of message routing approaches that are more energy-efficient [80], [197].

2.2.2.4. Processing

Among the analyzed papers it was possible to observe that initially, the main objective of IoT solutions was to collect and store data from sensor nodes. However, in more recent years, it is possible to observe an increasing number of IoT solutions that used supplementary techniques and technologies to treat the collected data, such as cloud computing and big data. Likewise, it is possible to observe an increasing number of works that used simultaneously two or more techniques or technologies for processing data. As shown in Figure 2.6, the most cited technologies within the reviewed papers are cloud computing (34%), machine learning (15%), big data (13%), and artificial intelligence (9%).

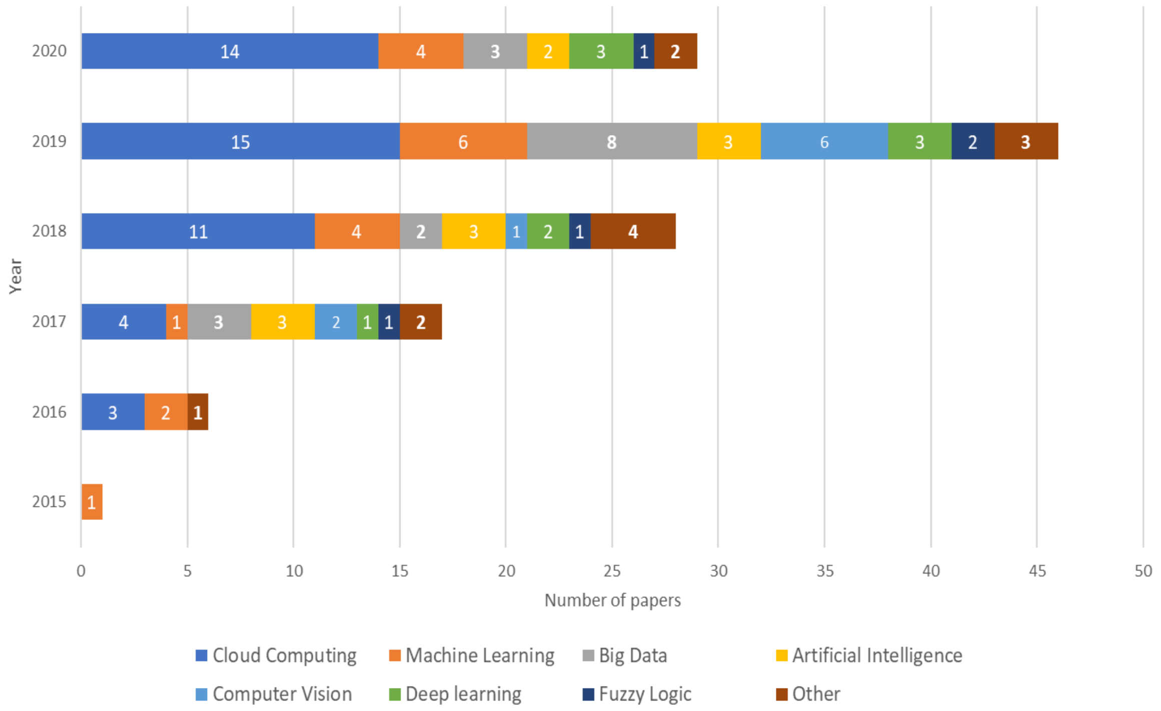


Figure 2.6 – Techniques and technologies for data-processing in smart farming identified within the reviewed papers.

Table 2.6 presents IoT solutions that relied on cloud-based platforms for processing data and highlights the main data processing techniques (e.g., artificial intelligence, big data, etc.). The column "Other/Not identified" comprehends IoT solutions that have used cloud-based platforms but have either (i) used any of the data processing technologies identified by other columns on Table 2.6 or (ii) not explicitly mentioned the type of data processing technology that was adopted.

Table 2.6 – IoT enabling platforms and data processing technologies used in smart farming within the reviewed papers.

Platform	Artificial Intelligence	Big Data	Machine Learning	Computer Vision	Other/Not Identified
AgroCloud		[198]			
AT&T M2X Cloud					[168]
AWS	[161]	[147]	[147], [161]		
Azure IoT Hub		[75]			[92]
Blynk					[144]
Cropinfra					[52]
Dropbox					[73], [97]
ERMES					[85]
FIWARE					[98], [100], [146], [148], [155]
Freeboard					[106]

Google			[119]		
GroveStream					[106]
MACQU					[129]
Mobius	[103]				
NETPIE					[110]
Rural IoT		[41]	[41]		[41]
Self-developed	[184]	[87]	[149]		[5], [72], [93], [122], [148], [199]
SmartFarmNET					[43]
Thinger.io		[44]			
ThingSpeak	[105]			[139]	[38], [45], [69], [88], [106], [112], [118], [123], [127], [130]
Ubidots					[39], [102], [125], [138]

Table 2.6 reveals that the most found cloud-based platforms in the reviewed papers are ThinkgSpeak, FIWARE, Ubidots, SmartFarmNet, AWS IoT and Thinger.io. In particular ThingSpeak is the most used cloud-based platform across all the reviewed papers, due to the fact that this platform is open-source with low infrastructure requirements [45]. In addition, Table 2.6 shows that AWS IoT was used with a higher number of data processing techniques. Not all cloud-based platforms offer the same set of functionalities, but in general, they have capabilities for data storage [41], [102], [110], [130], [139], processing [200] and visualization [110] and action control on farms [45]. Furthermore, Table 2.6 also reveals that, even though there are multiple cloud-based platforms, several reviewed papers developed their own cloud-based platform for the IoT solution.

Cloud-based platforms provide scalability for IoT solutions by relying on cloud computing to process and data. For instance, some platforms shown in Table 2.6, such as Thinger.io [44], are built entirely on top of infrastructure services provided by cloud providers (e.g., Amazon AWS and Microsoft Azure). Also supported by such services, the platforms make available data analysis modules with graphics and panels that allow real-time monitoring of the information obtained or the creation of customized panels from the integration of multiple data [44].

Due to the scalability provided by these platforms, the large amount of data generated by the sensors is stored in databases to form the so-called big data, an unstructured set of information that is used to generate information about crops. According to [201] big data demands the use of technologies to optimize the processing time due to the large volume of information. For example, Hadoop – a parallel database for big data applications – proved to be efficient when analyzing the rainfall index data from several meteorological stations [201].

IoT solutions use different types of techniques and technologies for processing the collected data. Table 2.7 presents commonly used technologies per applications as identified in the reviewed papers. Column "Other Technologies" encompasses all the technologies that are not identified by any of the other columns in Table 2.7.

Table 2.7 – Technologies and application in smart farming.

Application	Artificial Intelligence	Big Data	Computer Vision	Machine Learning	Blockchain	Fuzzy Logic	Other Technologies
Disease Prevention	[11], [141]		[136], [139]–[141]	[11], [140]			
Supply chain traceability		[169], [170]			[172], [175]		
Waste Management							[150]
Chemical control	[66]		[59]	[10], [59]			
Crop monitoring	[36], [103], [105], [133]	[41], [44], [75], [87], [121], [126], [133], [200]–[202]	[12]–[14], [84], [132], [133], [202]	[14], [36], [101], [119], [123], [126], [41], [58], [71], [81], [90], [91], [94], [95]	[41], [99]	[72], [81], [115], [128]	
Soil Management	[161], [165], [166]	[165], [166]		[161], [165]			
Vehicles and Machinery control							[52], [176]
Irrigation control	[57]	[57], [145], [147], [153], [157]		[57], [147], [149]		[159], [160]	[155]

Table 2.7 reveals that the most commonly used technologies to support data processing are artificial intelligence, machine learning, and big data. The use of these technologies is related to their ability to process large amounts of information in a short time. In addition, Table 2.7 also shows that crop monitoring is the most common type of application for IoT solutions that have used data processing technologies. Moreover, crop monitoring is also the type of

application that used the most different technologies for data processing. This can be understood by the fact that usually IoT solutions for monitoring crops collect a bunch of data and rely on machine learning and big data to process such data.

As demonstrated in Table 2.7, bigdata was used for different applications in IoT solutions, such as crop monitoring, soil management and irrigation control. For example, supported by big data, in [152], [166], [200] the soil moisture data gathered by physical sensors were related to data made available in datasets, such as the NASA Prediction of Worldwide Energy Resources (POWER) [203] – which contains meteorological data – purchase and sale values of crops, information from the user and government agencies to optimize the amount of water in irrigation cycles, support the farmer in the acquisition of agricultural inputs – such as seeds and fertilizers – and generate information and perspectives about other activities related to agriculture. Big data was also used by [66] in the development of a decision support system to provide irrigation and monitoring advice to farmers from a knowledge base created with data obtained by physical sensors (e.g., temperature, soil moisture) and virtual sensors (e.g., soil type, season). Virtual sensor is a type of software that, given the available information, processes what a physical sensor otherwise would [204].

In addition, automatic management with IoT depends on the manipulation of multiple variables. Initially, the simple observation of soil humidity and temperature can be used to trigger irrigation or cooling systems, as proposed by [191]. Nevertheless, greenhouse management can be more complex. As shown in [112], [115], [128], greenhouse parameters like temperature and humidity are closely tied and changing one of them can affect several others.

Fuzzy logic, as indicated in Table 2.7, was used in IoT solutions applications that need to handle multiple variables, such as irrigation control and monitoring crops. For this purpose, [128] uses fuzzy logic to handle multiple variables of temperature and humidity into a greenhouse and determine when a cooling system and an irrigation system should be started. Similarly, [81] uses fuzzy logic to optimize the number of sensors for monitoring soil temperature and moisture. Machine learning was also used in data processing by [57] to predict environmental conditions based on the forecast values of weather, humidity, temperature and water level and thus to control an irrigation system, by [58] to combine multiple parameters obtained from images, such as color and texture indices and by [59] to identify marks on the plants and, thus, to identify possible diseases. Similarly, in [13], [133] it was used to detect diseases, identify growth stages and the health of plantations.

Similarly, as shown in Table 2.7, IoT solutions used computer vision for applications that need to deal with image processing, such as crop monitoring and diseases prevention. It was also possible to observe in the reviewed papers the use of computer vision to identify and classify elements in images obtained by cameras, enabling the identification of fruit in an orchard [202] or the existence of diseases and pests in plantations [59], [136], [140]. Additionally, in [140] computer vision was used as a monitoring tool to detect the presence of insects that can cause diseases in olive groves and in [59] the same technique was employed to analyze diseases that cause morphological deformations in plants. Additionally, computer vision was used in crop management systems, for example in [141] where it was implanted in a robot equipped with a camera and other sensors, being able to obtain images of vegetation and, through computer vision, detect weeds in plantations and eliminate them. Similarly, in [119] a robot can identify a plant and interact with the environment to irrigate it, if necessary.

Finally, blockchain proved to be an opportune technology for systems that need to implement traceability of the supply chain, as shown in Table 2.7. According to [175] blockchain is a global public distributed ledger that records all transactions between users. In fact, this type of control is relevant for agriculture in several aspects, such as food safety, guarantee of origin or cost reduction. To ensure information security, this technology was proposed by [41], [172], [175] for agricultural product traceability. For example, in [175] an IoT solution uses blockchain to record information regarding the tea production based on 5 business processes: production plan, quality inspection, sales processing, product quality inspection and order delivery. In [172] a production tracking system for agricultural cooperatives have been developed. In [41] a similar system is being proposed but still in development stage.

2.2.3. Considerations

IoT solutions for smart farming take advantage of the scalability provided by platforms and cloud computing to store large amounts of data obtained by sensors. These big data of specific information may be processed with artificial intelligence techniques – such as machine learning – to improve the management of smart farming. For example, the processing of big data may be used to obtain crop insights, optimize water resources and increase the crop quality by preventing disease and reducing the amount of chemical products employed. Crop monitoring solutions use SBC (e.g., Arduino and Raspberry Pi) or UAV (e.g., drones) together with sensors (e.g., humidity, temperature, CO_2, or image) to collect data in indoor or outdoor environments.

Different types of network connections are used for communication between IoT devices, such as wired and wireless connections. The review shows that wired networks, such as CAN and Ethernet, are used for indoor agriculture (e.g., greenhouses). The use of wired network on indoor agriculture may be justified by the fact that in this scenario the physical components of the network are less susceptible to climatic agents impacts. Likewise, generally distance between sensor nodes in indoor agriculture enables this type of connection. Wireless connection, on the other hand, is used both in indoor and outdoor agriculture. Wi-Fi is the most mentioned protocol within the analyzed projects, due to its ubiquitous utilization in the daily life. However, power consumption and signal range characteristics may limit use of Wi-Fi in larger projects or in projects with power restrictions. To overcome the power consumption issue, energy-efficient protocols such as ZigBee, BLE or LoRa are used for communication in wireless networks.

Furthermore, it is worth mentioning that this review investigated papers where the IoT solution for smart farming was applied to agriculture only. However, the use of IoT for smart farming can also be applied to other activities related to farming, such as livestock [205]. Moreover, despite the fact that power-supply in IoT solutions for smart farming does not represent a specific layer of an IoT solution architecture [30], [206], this topic has been covered in some of the reviewed papers. For example, [80], [180] proposed improvements in algorithms for message routing and in duty cycles in sensor nodes. These approaches contribute to the reduction of power consumption by IoT devices. Similarly, a mission-based approach was used in [10] to optimize the power consumption in UAV. This approach was used to identify the most efficient path for a set of drones. Likewise, [180] proposed an intelligent activity cycle to improve the performance of data aggregators in terms of energy efficiency on cloudy days.

2.2.4. Conclusions

This work presented a systematic review of the state-of-the-art of IoT adoption in smart agriculture and identified the main components and applicability of IoT solutions. This review reported a change in the treatment of data in recent works: while previous work showed that the majority of decision support systems used simple processing mechanisms to handle data collected in real-time, more recent work showed an increasing number of management systems that use complementary technologies that rely on cloud and big data computing for processing large amounts of data. Furthermore, it was observed in this review that in recent work the use of artificial intelligence and image processing techniques has become more common to improve the management of smart farming. From the identified applications of IoT for smart farming it was observed that the most common application is the monitoring of crops. This review also showed that different network protocols may be simultaneously used in IoT solutions for smart farming. In addition, the comparison of types of network connections used in IoT solutions for smart farming revealed that wired networks are used in indoor scenarios (e.g., greenhouse) while wireless networks are used both in indoor and outdoor scenarios (e.g., arable lands, orchards). Moreover, the review discussed in this work suggests the increasing relevance of IoT solutions for smart farming. Future work may extend this review by including other relevant articles and complementary analysis of project costs, usability, and regional challenges intrinsic to IoT applications. Another important future research direction could be the analysis of the edge and fog computing usage in smart agriculture as a way to deal with challenges associated with traditional centralized cloud solutions such as high communication latencies, lack of support for real-time reaction to detected events, large bandwidths, etc.

2.3.Summary

Section 2 presented the theoretical framework, necessary for a better appreciation of the rest of this dissertation. This section was divided into two parts. The first part covered concepts, such as smart farming and IoT. It was discussed that smart farming is the use of supplementary technologies associated with agricultural production techniques in order to minimize waste and increase agricultural productivity. Likewise, section 2 discussed what IoT is, based on an architecture in 4 layers: perception, transport, processing, and application. Moreover, the characteristics of each layer of the architecture were discussed in detail. Based on the 4-layers IoT architecture, the second part of section 2 presented a systematic review of the state of the art of IoT solutions for smart farming. The systematic review aimed to identify the main techniques and technologies that enable IoT. For this purpose, the systematic review covered all the layers of the 4-layers IoT architecture. Based on the results provided by the systematic review, the architecture of an IoT solution capable of monitoring and acting in the mitigation of problems in plantations is proposed. This proposed architecture is based on the 4-layer IoT architecture, which consists of perception, network, processing, and application layers. The characteristics of the proposed architecture will be discussed in detail in section 3.

3. Proposed Architecture

As presented in section 2.2, several complementary techniques and technologies might be used in IoT solutions in smart farming scenarios. These techniques rely on cloud computing and big data to process large amount of data generated by the IoT solutions. The usage of big data and machine learning enable the processing of data in faster and efficient manner. Big data and machine learning also enable IoT systems for smart farming to work in a preventive manner, rather than simply processing collected data in a reactive manner.

Section 2.2 also shows that there are several IoT solutions for smart farming. Even though these solutions can solve specific problems, they cannot be easily applied for different scenarios. For example, in [11], the developed solution is hardly tied to the characteristics of the type of crop that is being applied, in this case, strawberries. Likewise, in [114] the number of sensors as well as their position within a greenhouse is fundamental for the proper functioning of the solution. Although these characteristics do not block the application of the solution to entirely different crops, they may increase the complexity of adapting the solution to different crops.

In contrast, the methodology of this work considered the above-mentioned characteristics and challenges to develop an architecture of an IoT solution that allows the use of machine learning and big data for processing data collected by sensor nodes. This architecture is suitable to be applied in the different agricultural environments (e.g., greenhouses, arable lands, orchards, etc.), as discussed in section 2.2.

3.1. System Overview

This section presents an architecture of an IoT solution capable of monitoring and acting in the mitigation of problems in plantations. The general functioning of this architecture is shown in Figure 3.1.

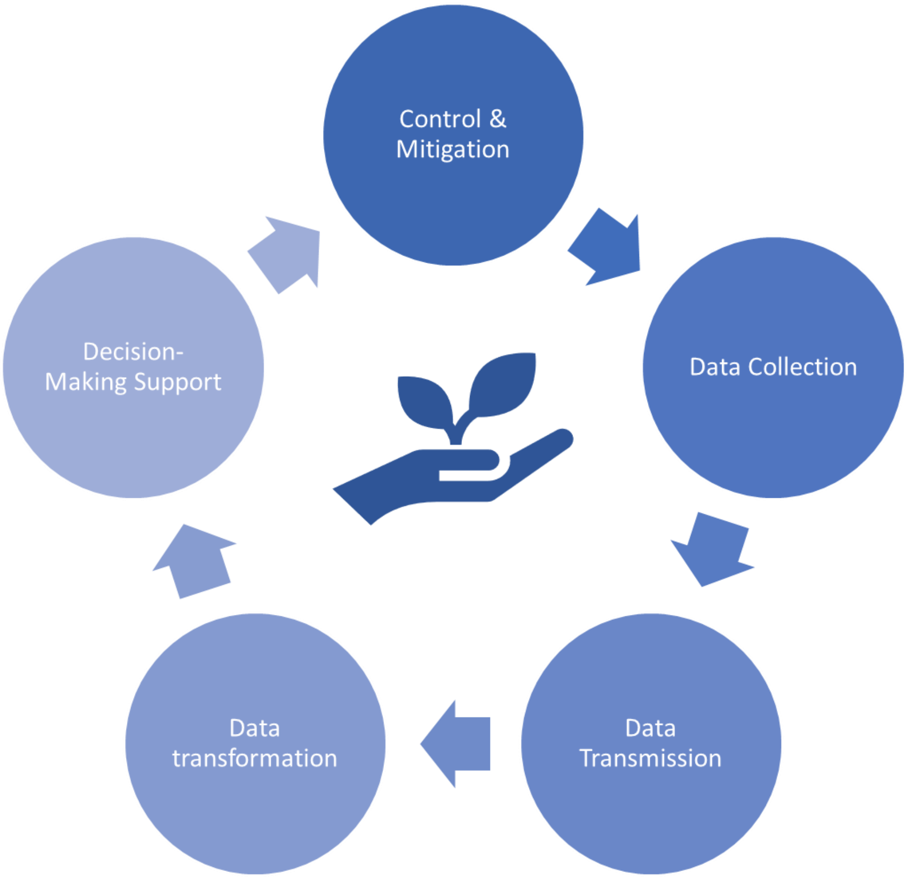


Figure 3.1 – Crops monitoring stages.

As illustrated in Figure 3.1, the raw data that are collected are transmitted to a remote system where they will be transformed in relevant information regarding the monitored crops. This information supports decision making and enables the control and mitigation of problems in crops.

3.2.Architecture Description

This proposed architecture is based on the 4-layers IoT architecture model presented in section 2.1.2, composed of perception, network, processing and application layers. Figure 3.2 presents the architecture diagram of the proposed architecture.

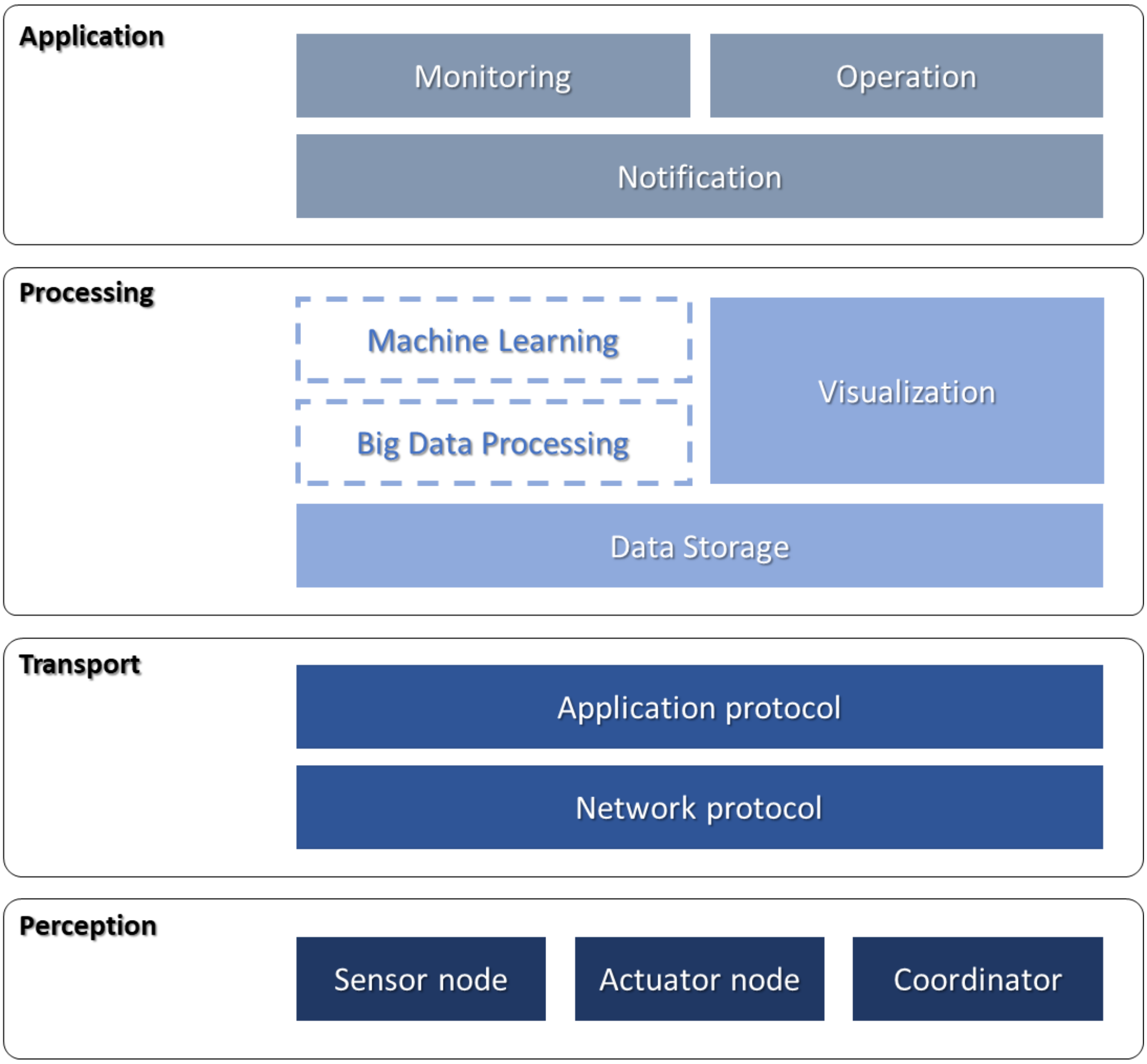


Figure 3.2 – Diagram of the proposed architecture.

As presented in Figure 3.2, the perception layer contains sensor nodes, actuator nodes and coordinator nodes. The sensor nodes are devices responsible for collecting data from the plantations. Multiple sensor nodes can be used simultaneously to collect different type of data from the crops, such as temperature, humidity, and luminosity. Actuator nodes are devices responsible for mitigating problems in the plantations. These actuator devices react to inputs from the remote platform that controls their status (e.g., running or stopped). The remote platform can trigger actuator devices automatically, based on information collected by the sensor nodes, or manually, based on manual inputs from the users. Coordinator nodes are responsible for intermediate the communication between sensor nodes and actuators with the application. The coordinator receives data transmitted for sensor nodes and retransmit such data directly to the remote platform or to another coordinator node. Moreover, the coordinator also routes messages from the remote platform to the actuators.

Sensor nodes, actuators and coordinators can be logically grouped in monitoring clusters. Data collected by a cluster are stored in a structured way, so that this data does not interfere with the data of the other clusters, as illustrated in Figure 3.3.

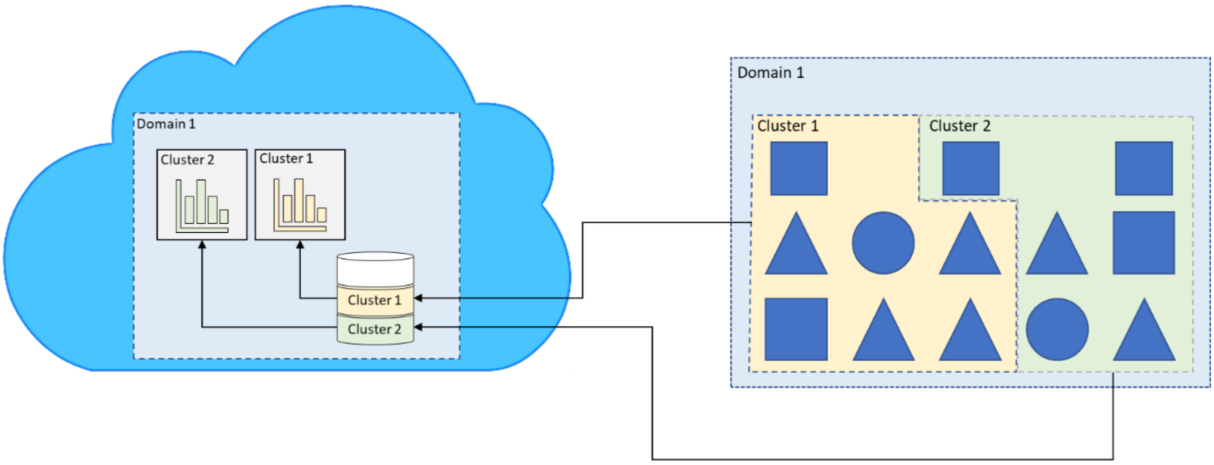


Figure 3.3 – Monitoring clusters.

As shown in Figure 3.3, data collected by a given cluster are stored in a database separately from data collected by other clusters. Such separation allows, for example, the simultaneous monitoring of different types of crops, or the sectorization of extensive plantations into smaller monitoring clusters. Nevertheless, all clusters are grouped into a logical unit called domain. The domain contains information common to the different clusters, such as farm data or information about the users of a IoT solution that implements the proposed architecture.

The transport layer contemplates communication technologies used by sensor nodes, actuators, coordinators, and the remote platform. As discussed in section 2.2, different types of network technologies can be used by sensor nodes and actuators, such as Wi-Fi, Bluetooth, Zigbee, etc. Due to this fact, in the proposed architecture the coordinator is responsible for equalizing the different communication technologies and transmitting data consistently to the remote platform, as illustrated in Figure 3.4.

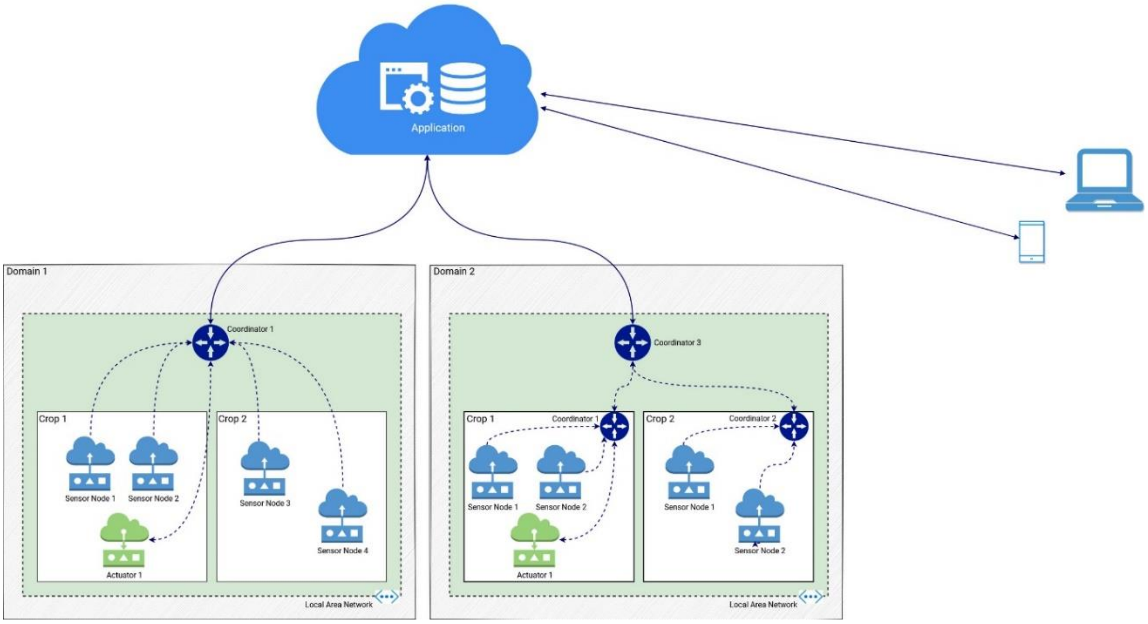


Figure 3.4 – Network Architecture Diagram.

As shown Figure 3.4, the proposed architecture considers the existence of multiple domains and crops. In Figure 3.4, each domain represents a farm where multiple crops can exist. Several sensor nodes and actuators can be associated to a particular crop. These sensor nodes and actuators send messages to a coordinator node via a short or mid-range network technology (e.g., Wi-Fi, Bluetooth, etc.). The coordinator node can forward messages directly to the remote application via Internet or to another coordinator node within the same domain.

The message exchange in the proposed architecture follows the publisher/subscriber architecture. As discussed in section 2.2.2, the publisher/subscriber architecture is suitable for devices with limited computing resources. Moreover, as shown in Figure 3.5, the publisher/subscriber architecture can improve scalability by enabling low coupling between the publishers and the subscribers.

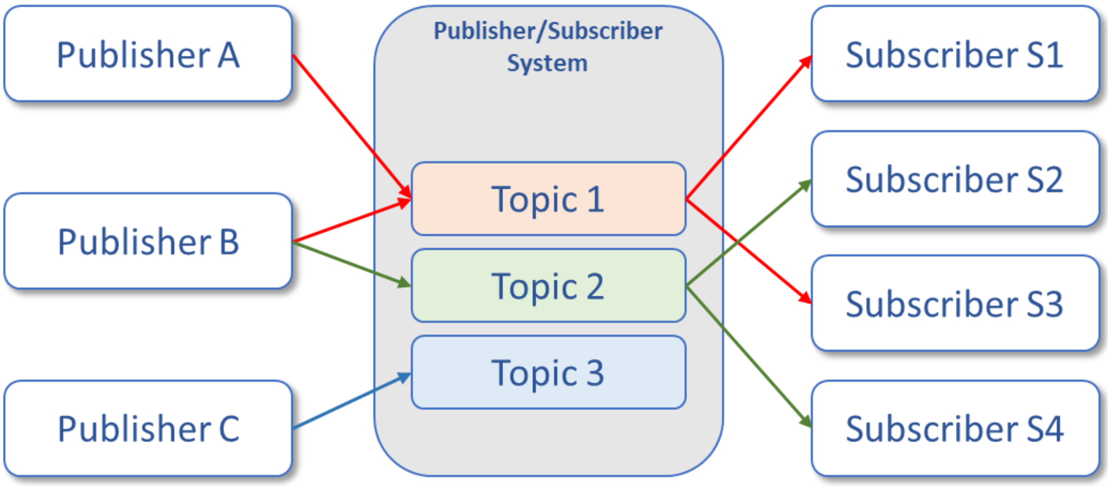


Figure 3.5 – Publisher/subscriber architecture.

As shown in Figure 3.5, both Publisher A and Publisher B can publish messages to a topic named Topic 1. Additionally, Publisher B also publishes messages to a topic named Topic 2 while Publisher C publishes messages to a topic name Topic 3. Complementary, Figure 3.5 shows that two subscribers (S1 and S3) listen to the messages published on Topic 1, another two subscribers (S2 and S4) listen to the messages published on Topic 2 and no subscribers listen to the messages published on Topic 3. Therefore, Figure 3.5 illustrates that the publisher/subscriber architecture enables devices to publish messages in topics regardless of whether any device is listening those messages. In the same way, a device can subscribe to one or more topics regardless of any messages being published to those topics. Moreover, multiple devices can publish messages or subscribe to the same topic. This behavior provides a simple way to add or to remove sensor nodes in our architecture. In the proposed architecture it is expected that some devices can act both as publisher and subscribers at the same time, as presented in Table 3.1.

Table 3.1 – Communication roles.

Device	Publisher	Subscriber
Sensor Nodes	Yes	No
Actuators	Yes	Yes
Coordinators	Yes	Yes

As shown in Table 3.1, sensor nodes, actuators and coordinators can act as publishers by publishing messages to specifics topics. For example, sensor nodes can publish temperature and humidity data and actuators can publish their own status (e.g., running or stopped).

Analogously, actuators and coordinators can subscribe to one or more topics and listen messages published by other components of the proposed architecture. For example, coordinators can subscribe to topics where sensor nodes publish temperature data and forward this information to the remote application. Likewise, actuators can subscribe to topics to listen the messages that control their status.

Figure 3.2 shows that the processing layer contains the data storage and data visualization modules. In addition, the processing layer allows the use of machine learning and big data processing on the stored data. These modules rely on cloud-computing to provide scalable storage and processing power for extensive amounts of data collected by sensor nodes. The data storage module stores all the information about the other modules all layers of the proposed architecture. This information can include information about the domain (e.g., farm, crops, and users), devices (e.g., sensor nodes, actuators, coordinators) and crops (i.e. data collected by sensor nodes). The data visualization module enables the use of tools that are used to present information generated by other modules to the users (e.g., dashboards, charts, etc.). Furthermore, the architecture allows the usage of big data and machine learning modules to. The big data module allows the usage of big data to provide fast extraction of information from the data collected by sensor nodes. The machine learning module allows the usage of machine learning to processes the information stored in the data storage module.

Finally, as shown in Figure 3.2, the application layer includes monitoring, operation, and notification bus modules. In the proposed architecture, the monitoring module is responsible for observing whether the metrics collected by the sensor nodes are within defined limits for a given cluster. The operation module controls the state of actuators deployed in the cluster. The notification bus supports the exchange of messages between the monitoring module and the operation module. For example, if the metrics observed by the monitoring module are outside the limits defined for the cluster, the monitoring module notifies the operation module via the notification bus.

3.3.Summary

This section presented the proposed architecture of an IoT solution capable of monitoring and acting in the mitigation of problems in plantations. The architecture proposed in this work encompasses several concepts that enable its usage in different contexts of agriculture, i.e. outdoor and indoor (section 2.2.2), and in plantations of different types and extensions. This proposed architecture is based on the 4-layer IoT architecture model (section 2.1.2), composed of perception, network, processing and application layers. The perception layer includes devices responsible for collecting data and mitigating problems in plantations (sensor nodes and actuators). This layer also includes coordinator nodes, devices that are responsible for intermediate the communication between the other devices with the remote platform. In this layer it was presented that sensor nodes and coordinators can be logically grouped in domains and clusters. It was also presented that such organization enables the simultaneous monitoring of different types of crops, or the sectorization of extensive plantations into smaller monitoring clusters. The transport layer comprises the communication technologies used by devices of the perception layer to communicate with the remote platform. In this layer it was detailed how sensor nodes and actuators communicate with coordinator nodes and how sensor nodes communicate with the platform or with other coordinator nodes. It was also explained that the message exchange in the proposed architecture follows the publisher/subscriber architecture and it was detailed the communication roles (i.e., publisher or subscriber) assumed by the devices of the perception layer (i.e., sensor nodes, actuators and coordinators). The processing layer contains the data storage and data visualization modules. In this layer the structure of the stored data (domain, devices, crops) was discussed. It was also discussed that processing layer allows the use of machine learning and big data processing on the stored data. Finally, the application layer comprises the monitoring, operation, notification bus modules. In this section, the functions of the monitoring, operation, and notification modules were covered. It was discussed that the monitoring module is responsible for verifying if the data collected by sensory nodes are outside a defined limit and notify the operation module through the notification module. It was also discussed that the operation module is responsible for controlling the status of actuators in the plantations.

The next section presents a prototype that has been implemented to validate the proposed architecture discussed in this section.

4. Prototype Implementation

This chapter is divided in three parts: (i) implementation and (ii) tests of a prototype to validate the architecture discussed in section 3, and (iii) security risks evaluation.

4.1. Hardware Components

In order to validate the functioning of the proposed architecture in section 3, a prototype that implements the main components of the architecture has been developed. In this prototype both sensor nodes and actuators are implemented with the ESP8266 NodeMCU Wi-Fi Devkit. This board has an integrated wireless connection through the ESP8266 Wi-Fi SOC (Espressif Systems CO LTD, Shanghai, China), a storage unity called SPIFFS and a set of programmable input and output pins to which sensors and actuators, among other things, can be connected as shown in Figure 4.1 [207].

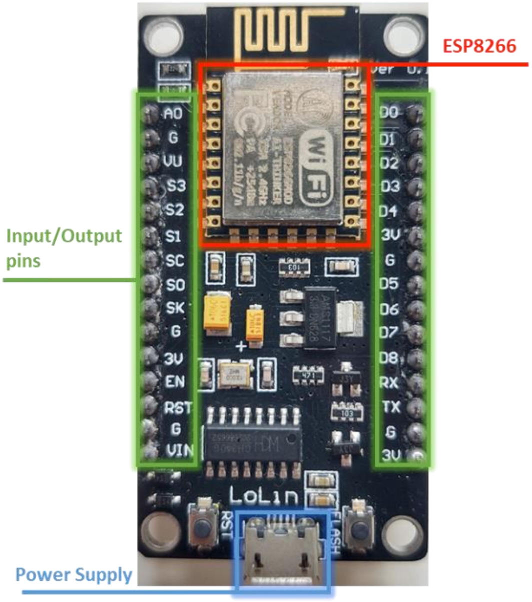


Figure 4.1 – ESP8266 NodeMCU.

According to [208], ESP8266 NodeMCU is a component developed specifically for mobile devices and IoT applications that has low power consumption thanks to proprietary technologies and architectures implemented by the developer. Table 4.1 presents the main characteristics of this component.

Table 4.1 – Specifications of ESP8266 NodeMCU [208].

Parameter	Value
Protocols	802.11 b/g/n
Frequency	2.5GHz
Operating Voltage	2.5V ~ 3.6V
Operating Current	80mA
Operating Temperature	~-40°C - 125°C
Storage Temperature	~-40°C - 125°C

ESP8266 NodeMCU supports execution of self-developed firmware to customize the behavior of the board. Several tools and programming languages can be used to developed custom firmware for ESP8266 NodeMCU. For the development of the custom firmware in this prototype it was used the Arduino IDE[1], which is a tool that allows the development, testing and deployment of custom firmware for the NodeMCU with C/C++ [188].

Additionally, for the sensor nodes it was used the temperature sensor DS18B20 (Maxim Integrated, San Jose, CA, USA) and the moisture sensor YL-69. Sensor DS18B20 is a digital thermometer that provides temperature in Celsius degrees with a precision of up to 12 bits [209]. This sensor is particularly suitable to be used with the NodeMCU because, as shown in Figure 4.2, communication takes place through a single bus, which means that this sensor can obtain energy and data from one single connection. Thanks to this characteristic, it is possible to obtain the desired measurements using the minimum number of ports of a NodeMCU.

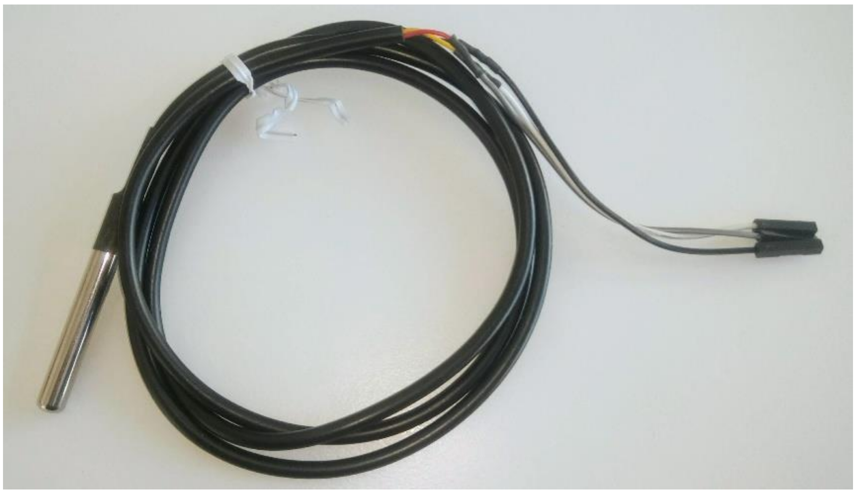


Figure 4.2 – Temperature sensor DS18B20.

[1] Arduino IDE is available in: https://www.arduino.cc/en/Main/Software.

Sensor YL-69 is used to measure the moisture of the soil. As shown in Figure 4.3, this sensor is composed of two probes that are responsible for measuring the electrical resistance in the substrate.

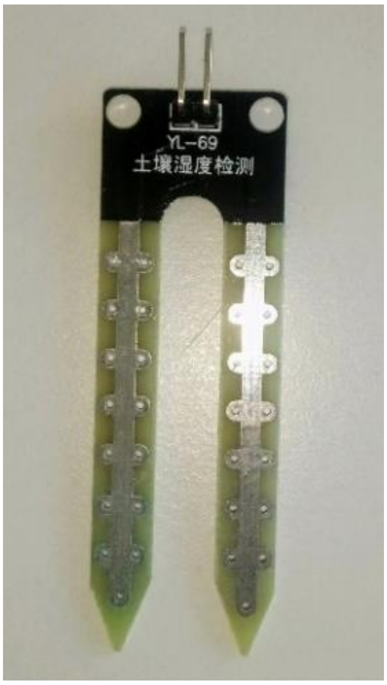


Figure 4.3 – Moisture sensor YL-69.

The output signal varies between 0V and +4.2V depending on the amount of water present in the substrate. The greater the amount of water in the substrate, the lower the electrical resistance. The measured signal is sent to a comparator LM393 (Texas Instruments, Dallas, TX, USA) that converts electrical signals to an analog value that the NodeMCU can process. Values obtained from the sensor may vary from 0 to 1023. As indicated in Table 4.2, the lower values indicate a wetter soil and the higher values indicate a drier soil.

Table 4.2 – Sensor Reading and Soil Condition [210].

Sensor Reading	Soil Condition
0-600	Wet soil
601-950	Moist soil
951-1023	Dry soil

The sensors specifications are presented in Table 4.3.

Table 4.3 – Temperature sensor and moisture sensor specifications, based on [209], [211]

Sensor	Input voltage	Input current	Output voltage	Temperature
Moisture	+3.3V - +5V	35mA	0V - 4,2V	-10°C - +50°C
Temperature	+3V - +5.5V	1mA	+3V - +5.5V	-55°C - +125°C

As shown in Table 4.3, sensor DS18B20 operates in temperatures between -55°C and 125°C and YL-69 sensor operates in temperatures between -10°C and +30°C. In temperatures between -10°C and +85°C – which is a reasonable scenario for agricultural environments – sensor DS18B20 has an error of ±0,5°C [209]. Moreover, both sensors DS18B20 and YL-69 need a power supply between +3V and +5,5V. This characteristic enables these sensors to be connected directly to the output terminals of the NodeMCU without an external power supply.

Regarding the actuators, to simplify the development of this prototype, a three segment light-emitting diode (LED), like the one shown in Figure 4.4, is used to emulate the actuators status.

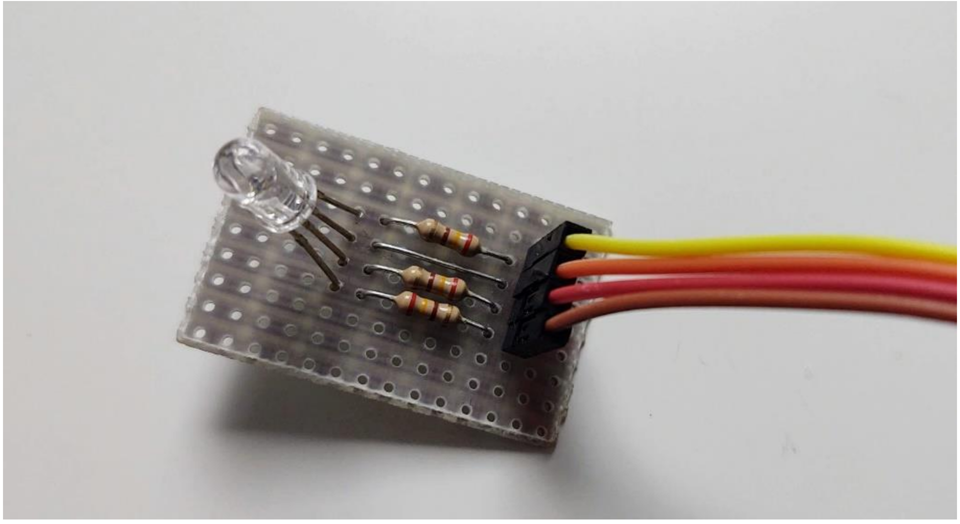


Figure 4.4 – Three segment LED.

This electronic component can emit light in different colors depending on which segment of the LED is powered. The LED is connected to the NodeMCU ESP8266, so that the LED colors can indicate whether an actuator is running or stopped as described in Table 4.4.

Table 4.4 – Correlation between LED color and status of actuators

LED Status	Description
Off	Both actuators are stopped
Blue Light	Irrigation actuator is running
Red Light	Fan actuator is running

A customized firmware has been developed by the author specifically for the NodeMCU. This firmware aims to allow that sensor nodes can connect and transmit data to the coordinator nodes as well as the actuators can receive instructions from the remote platform through the coordinator node. The basic workflow of the firmware is presented in Figure 4.5.

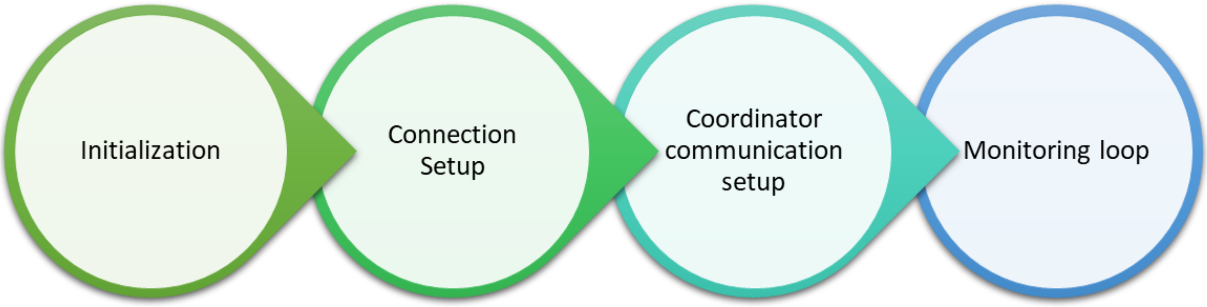


Figure 4.5 – NodeMCU firmware workflow.

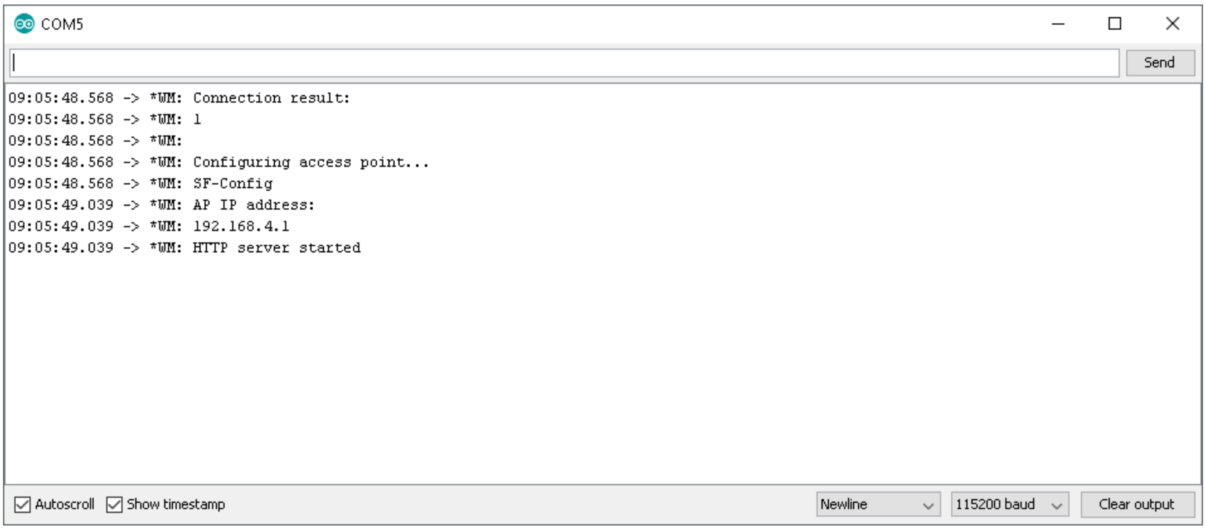
As indicated in Figure 4.5, during the initialization step the program defines control variables and load the required libraries for the operation of the firmware and for the reading of the sensors.

Table 4.5 presents the main libraries used by the firmware and their purpose for the application.

Table 4.5 – Libraries used in the firmware.

Library	Purpose
ArduinoJson 5.13.4	Process messages in JSON format
DallasTemperature 2.3.4	Interface with the temperature sensor
DNSServer	Runtime network configuration
ESP8266WebServer	Runtime network configuration
ESP8266WiFi	Wireless connection manager
FS	Runtime parameters definition
OneWire 3.8.0	Interface with the temperature sensor
PubSubClient 2.7.0	Connection to the coordinator node using MQTT protocol
WiFiManager 0.14.0	Runtime network configuration

During the connection setup step, the program search for a network to connect the device. If there is no valid network (for example, in the case of the first initialization of a sensor node), the program starts a configuration mode. In the configuration mode, the program makes available a configuration page from where it is possible to insert network and coordinator settings, as shown in Figure 4.6 and Figure 4.7. Once the settings are validated, they are persistently stored in the SPIFFS module within the NodeMCU. Thus, even if the device is turned off the same connection settings will be used on the next boot.



```
09:05:48.568 -> *WM: Connection result:
09:05:48.568 -> *WM: 1
09:05:48.568 -> *WM:
09:05:48.568 -> *WM: Configuring access point...
09:05:48.568 -> *WM: SF-Config
09:05:49.039 -> *WM: AP IP address:
09:05:49.039 -> *WM: 192.168.4.1
09:05:49.039 -> *WM: HTTP server started
```

Figure 4.6 – NodeMCU serial console log.

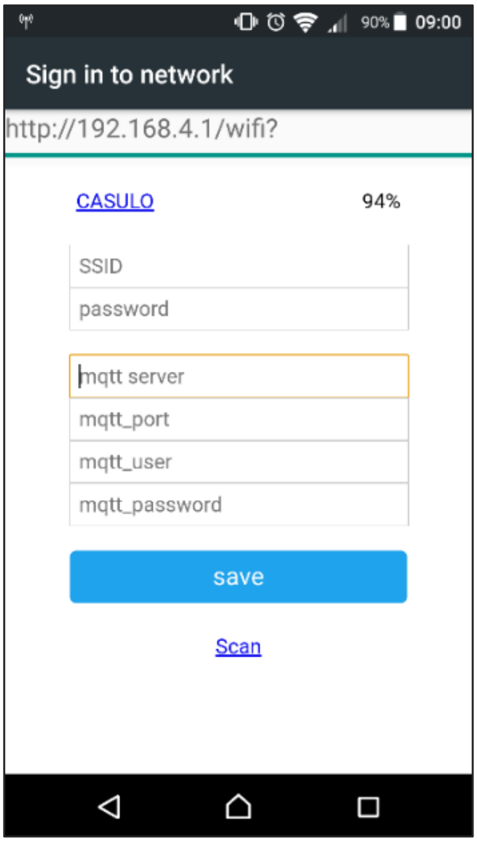


Figure 4.7 – Configuration mode screen.

During the coordinator communication setup step, the program connects the sensor node or actuator to the coordinator using the settings defined in the step before. If the connection is successful, the program enters in the monitoring loop step. The monitoring loop step can work in different two ways: for sensor nodes, the program checks and transmits data collected by sensors to the coordinator node. For the actuator nodes the program checks for messages from the remote platform on the coordinator node. Moreover, to save energy in the sensor nodes and actuators, a mechanism wake-up/sleep cycle was implemented in the code.

Coordinator nodes are implemented in this protype with a Raspberry Pi (Raspberry Pi Foundation, Cambridge, UK). Raspberry Pi is a low-cost SBC with small dimensions and high computing power. A Raspberry Pi is composed of microprocessor, RAM, video processing unit, storage and a set of input and output terminals called GPIO. Some models also have embedded network modules to provide connection via Bluetooth or Wi-Fi, for example, as show in Table 4.6.

Table 4.6 – Raspberry Pi, based on [189], [212].

Version Model	**CPU**			**RAM**	**GPU**	**Connectivity**		
	Chipset	**Clock Speed**	**Core**			**Wi-Fi**	**Bluetooth**	**GPIO**
Raspberry Pi Zero	BCM2835	1GHz	Single Core	512 MB	-	-	-	
Raspberry Pi Zero W	BCM2835	1GHz	Single Core	512 MB	-	802.11 b/g/n	4.1	
Raspberry Pi 1 A+	BCM2835	700MHz	Single Core	256 MB	Videocore IV @ 400Mhz	-	No	40
Raspberry Pi 1 B+	BCM2835	700MHz	Single Core	512 MB	-	-	No	40
Raspberry Pi 2 Model B	BCM2837	900MHz	Quad Core	1 GB	Videocore IV @ 400Mhz		No	40
Raspberry Pi 3 Model B	BCM2837	1.2GHz	Quad Core	1 GB	Videocore IV @ 400Mhz	802.11 b/g/n	4.1	40
Raspberry Pi 3 Model B+	BCM2837B0	1.2 GHz	Quad Core	1GB	Videocore IV @ 400Mhz	802.11 b/g/n/ac	4.2	40

Raspberry Pi 4 Model B	BCM2711	1.5 GHz	Quad Core	1GB / 2GB / 4GB	Videocore VI @ 500Mhz	802.11 b/g/n/ac	5	40

As shown in Table 4.6, Raspberry Pi models differ from each other in terms of the amount of RAM, processing power and communication technologies. Moreover, as discussed in section 2.2.2.2, Raspberry Pi uses a high-level OS to manage peripherals and operations, (e.g., Raspbian, Ubuntu Core, etc.). In general, these OS support applications developed with sophisticated programming languages such as Python. The model used in this prototype (Raspberry Pi 3 Model B) offers 1GB RAM with embedded Wi-Fi and Bluetooth (Table 4.6). These characteristics enables the Raspberry Pi to execute software components that constitute the IoT solution.

In order to operate as a coordinator node, an application that implements communication using MQTT protocol (MQTT broker) and an application that were developed by the author were installed on the Raspberry Pi. As presented in section 2.2, MQTT is a protocol that implements publisher/subscriber architecture to enable communication between hardware-constrained devices or devices in networks limited by bandwidth and high latency. In this communication architecture multiple clients can subscribe to or publish messages on one or multiple topics, as discussed in section 2.1.2. Moreover, MQTT has native security mechanisms, such as authentication between the server and the clients and topics filtering, which allows restricting clients that can listen to specific topics. Such security mechanism is used in the prototype to implement authentication between the MQTT server installed in the coordinator nodes and the sensor nodes and actuators to ensure only authorized devices can receive or publish messages to the solution. The application developed by the author aims to route messages received both from the sensor nodes and platform. Considering that the amount of information to be processed by the coordinator node can be large, the application was developed in Python because, according to [213], Python is an appropriate programming language to handle large volumes of data. The basic workflow of the application developed by the author is presented in Figure 4.8.

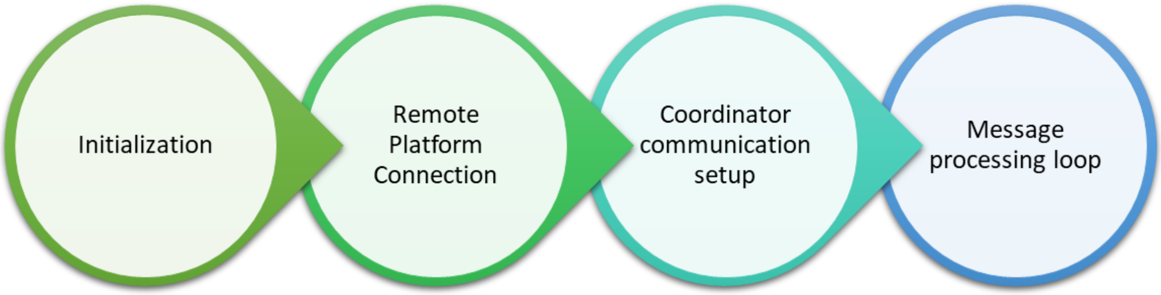


Figure 4.8 – Workflow of the application developed to route messages between the sensor nodes and platform.

As shown in Figure 4.8, during the initialization step the self-developed application loads all the libraries and set variables required for the operation of the application. In the remote platform connection step, the self-developed application attempts to connect to the remote platform by using pre-defined credentials. Then, in the coordinator communication setup step, the application connects to the MQTT server and subscribes to several topics. Finally, the application enters in the message processing loop step. In this step the application listens to messages that are published in the MQTT. When a new message arrives in MQTT, the application appends metadata to the message (e.g., timestamp and coordinator node identification) and forward the message to the remote platform. Moreover, the application also listens to messages that came directly from the platform. When the application receives a new message from the remote platform, the application publishes this message to one MQTT topic, so that the actuators can receive the message. Figure 4.9 shows the connection of the components in the prototype.

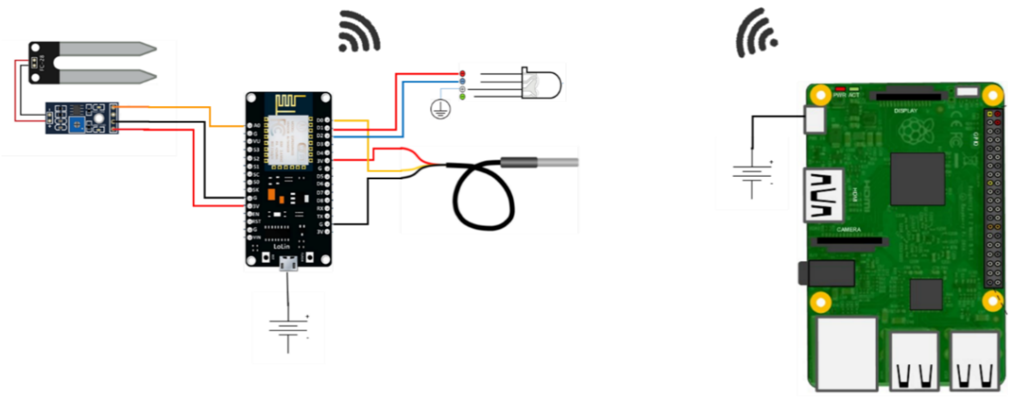
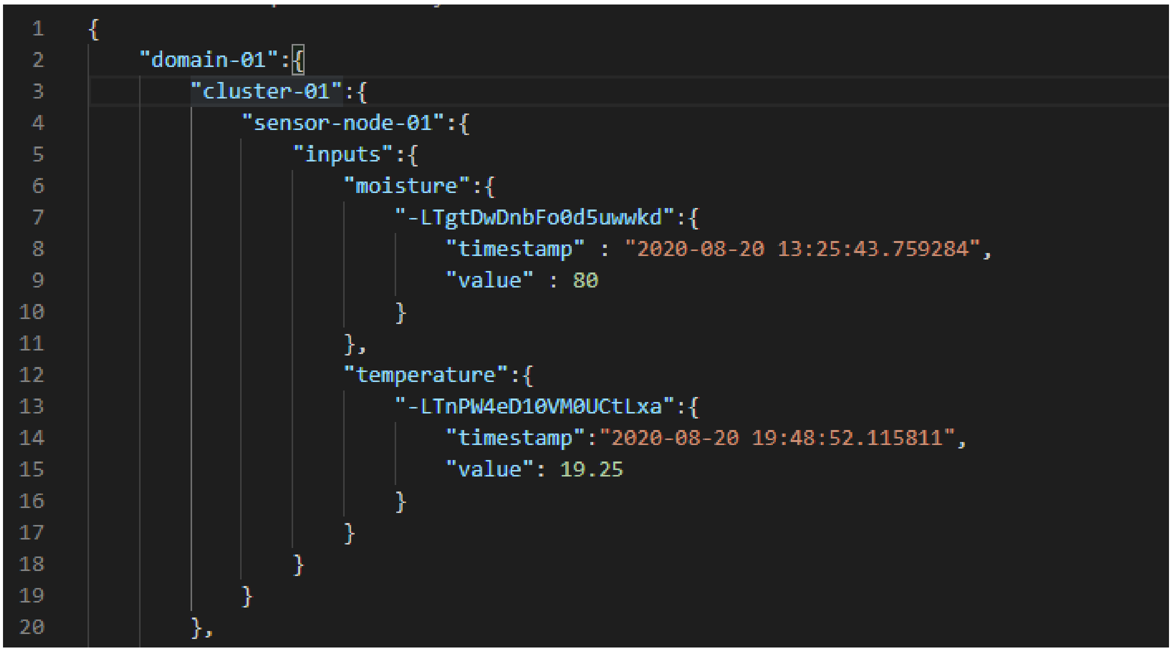


Figure 4.9 – Connection of the components in the prototype.

The remote platform in this prototype contains the database and the web application. The database used in this prototype is Firebase Realtime Database. The Firebase Realtime Database is a cloud-hosted NoSQL database that allows to store and synchronize information among several devices. Differently from common relational databases, in a NoSQL database the information is stored in a JSON structure [214]. The database stores information about the domain, clusters, devices, and data collected by sensor nodes of the prototype. Data collected by sensor nodes that belong to a given cluster are stored in a structured manner to ensure isolation between data collected by sensor nodes that belong to other clusters, as shown in Figure 4.10.



```
{
    "domain-01":{
        "cluster-01":{
            "sensor-node-01":{
                "inputs":{
                    "moisture":{
                        "-LTgtDwDnbFo0d5uwwkd":{
                            "timestamp" : "2020-08-20 13:25:43.759284",
                            "value" : 80
                        }
                    },
                    "temperature":{
                        "-LTnPW4eD10VM0UCtLxa":{
                            "timestamp":"2020-08-20 19:48:52.115811",
                            "value": 19.25
                        }
                    }
                }
            }
        },
```

Figure 4.10 – Example of how data collected by sensor nodes are stored in the database.

Figure 4.10 shows an example of data collected by a sensor node named “sensor-node-01” that are stored under a domain named “domain-01” and under a cluster named “cluster-01”. The structure presented in Figure 4.10 allows a domain to support multiple clusters and each cluster to store data from multiple sensor nodes (section 3.2). The architecture implemented in this prototype allows the use of big data and machine learning tools to process the data stored in the Firebase Realtime Database (Figure 4.10).

A web application has been developed to allow the user to interact with the solution at any time and anywhere. To do this, the web application presents in real time information stored in the database. In addition, mechanisms have been implemented to allow the user to configure the monitoring thresholds for temperature and humidity. The monitoring thresholds are used by the operating module (section 3.2) to define alerts and the status of the actuators. The main screen of the web application is divided into three sections, as indicated in Figure 4.11, Figure 4.12 and Figure 4.13. The first section presents a panel with the system status and updated information in real time. The second section presents controls that allow the user to interact with the actuators manually. Finally, the third section presents controls that allow the user to define the maximum and minimum monitoring thresholds for temperature and humidity that will be used later by the operation module (section 3.2) to automatically activate the actuators.

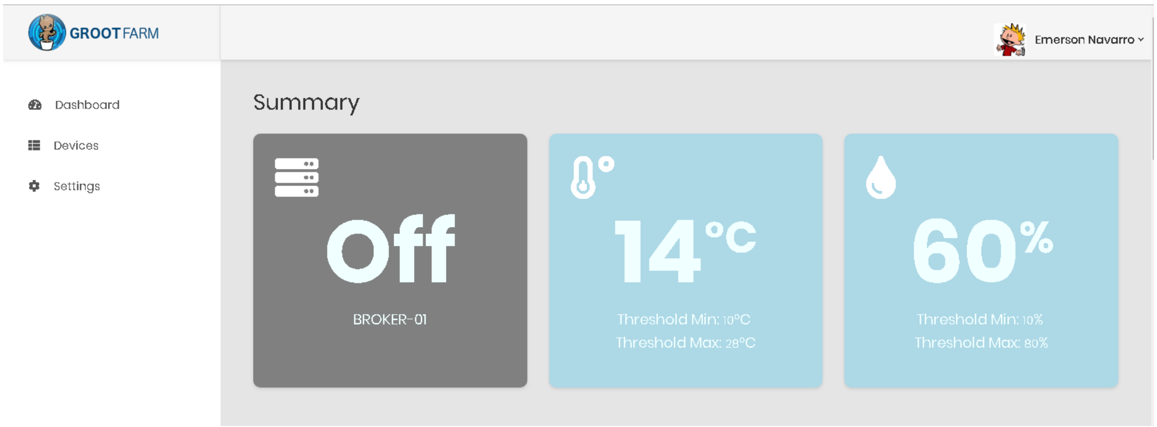


Figure 4.11 – Web application - Section 1: Panel.

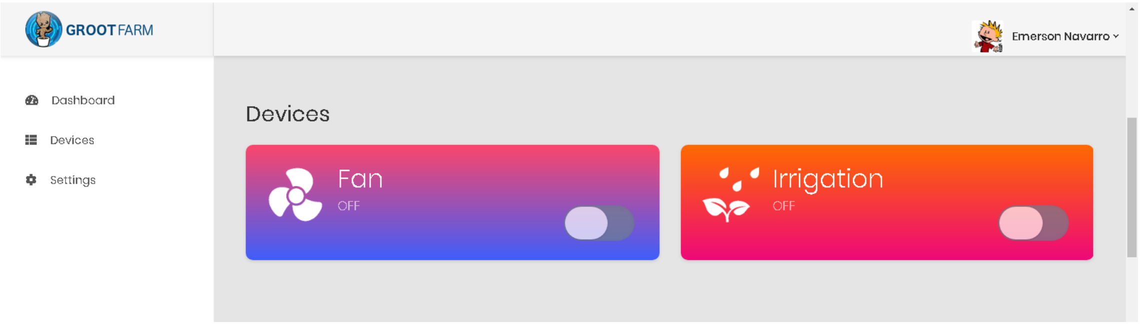
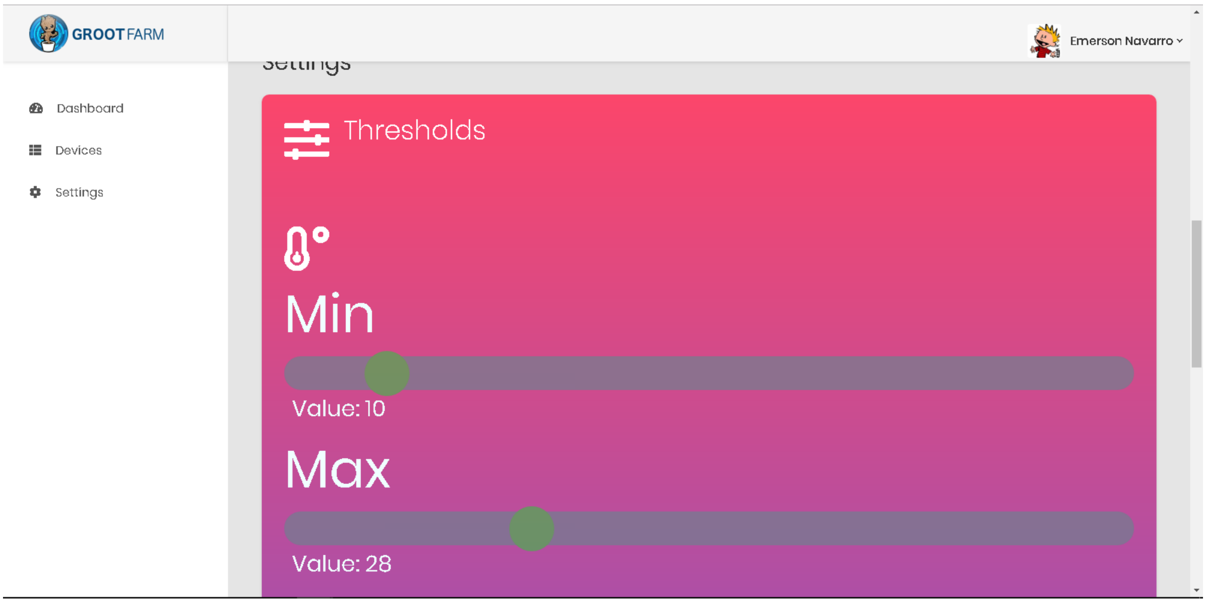


Figure 4.12 – Web application - Section 2: Actuators Control.



Min
Value: 10

Max
Value: 28

Figure 4.13 – Web application - Section 3: Thresholds settings.

Finally, responsive technologies and standards were adopted during the development of the web application to improve the user experience in internet-devices with different screen sizes, as shown in Figure 4.14.

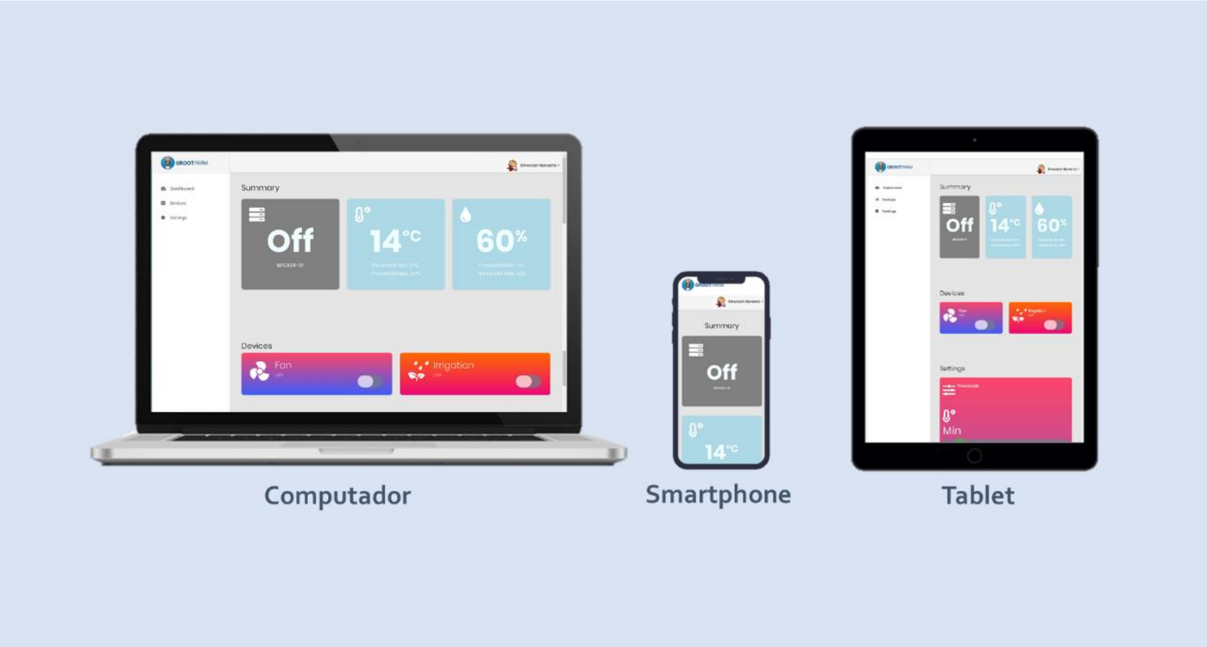


Figure 4.14 – Web application developed with responsive technologies and standards.

4.2.Tests and Validation

Tests are divided into two phases: operational tests and functional tests. Operational tests aimed to determine whether all components work and can communicate with each other. Thus, tests performed in this phase include:

- Functioning tests on each device used in the prototype (sensors, light-emitting diode, NodeMCU, and Raspberry Pi).
- Validation of the type of data collected by the sensors.
- Connectivity tests between the NodeMCU and the MQTT server.
- Connectivity tests between the Raspberry Pi and the database.

Functional testes aimed to validate the implementation of the architecture discussed in section 3 through the operation of the prototype. Table 4.7 presents the set of test cases created for this purpose.

Table 4.7 – Test cases.

Case	Name	Description	Expected Result
1	Credentials validation	Verify the behavior of the sensor node when a user provides invalid credentials during the initial configuration of the sensor node.	The sensor node does not connect to the network; The "configuration mode" remains enabled in the sensor node.
2	Sensor node connection	Verify the behavior of the sensor node when a user provides valid credentials during the initial configuration of the sensor node.	The sensor node connects to the network; The sensor node disables the "configuration mode".
3	Sensor nodes – persistent configuration	Verify the behavior of the sensor node when the sensor node has a valid network configuration.	The sensor node connects to the network with the last valid configuration. The "configuration mode" remains disabled.
4	Data collection – temperature	Verify that temperature changes are detectable.	The web application dashboard displays new temperature values.
5	Data collection – moisture	Verify that soil moisture changes are detectable.	The web application dashboard displays new soil moisture values.
6	Fan actuator operation – high temperatures	Verify the operation of the fan actuator when temperature is greater than the maximum temperature threshold	The web application dashboard displays the new temperature, and the fan actuator is started.

7	Fan actuator operation – low temperatures	Verify the operation of the fan actuator when the measured temperature is lower than the minimum temperature threshold.	The web application dashboard displays the new temperature, and the fan actuator is stopped.
8	Irrigation actuator – dry soil	Verify the operation of the irrigation actuator when the value measured for soil moisture is lower than the minimum moisture threshold	The web application dashboard displays the new soil moisture value, and the irrigation actuator is started.
9	Irrigation actuator – wet soil	Verify the operation of the irrigation actuator when the measured moisture value is higher than the maximum moisture threshold.	The web application dashboard displays the new moisture value, and the moisture irrigation actuator is stopped.
10	Update temperature threshold	Verify the behavior of the system when the maximum temperature threshold increases.	The new maximum temperature threshold is displayed in the threshold settings section of the web application (Figure 4.13). When the measured temperature is above the new maximum temperature threshold the fan actuator is started.
12	Update moisture threshold	Verify the behavior of the system when the minimum moisture threshold decreases.	The new minimum moisture threshold is displayed in the threshold settings section of the web application (Figure 4.13). When the measured moisture is below the new minimum moisture threshold the irrigation actuator is started.
11	Manual interaction validation	Verify the behavior of the system when the user interacts with the actuators manually from the console.	The status of actuators is updated in the actuators control section of the web application. Actuators are started or stopped.

The results of the tests are presented in Table 4.8.

Table 4.8 - Results of functional tests.

Case	Name	Performed steps	Obtained results	Conclusion
1	Credentials validation	The sensor node has been started. It was verified that the configuration mode was activated. The user provided invalid credentials in the “configuration mode” page.	The sensor node did not connect to the network and when restarting the configuration mode remained active (Figure 4.7).	Satisfactory

2	Sensor node connection	The sensor node has been started. It was verified that the configuration mode was activated. The provided valid credentials in the "configuration mode" page.	The sensorial node has connected to the network and the configuration mode has been deactivated.	Satisfactory
3	Sensor nodes – persistent configuration	The power supply of the sensor node has been removed and reinserted.	The sensor node has connected to the last valid network and the configuration mode has not been activated.	Satisfactory
4	Data collection – temperature	The temperature sensor was positioned near a heat source to induce the increase of temperature. The temperature increased from 19°C to 32°C degrees.	The new temperature (32°C) was displayed on the web application dashboard correctly (Figure 4.11).	Satisfactory
5	Data collection – moisture	The moisture sensor was positioned near a humidity source to induce the increase of humidity. The humidity increased from 45 to 99.	The new moisture value (99) was displayed on the web application dashboard correctly (Figure 4.11).	Satisfactory
6	Fan actuator operation – high temperatures	The maximum temperature threshold is set to 30°C (Figure 4.13). The temperature sensor was positioned near a heat source to induce the temperature increase until the measured temperature was higher than 30°C.	The new temperature was displayed correctly in the web application dashboard (Figure 4.11). When the measured temperature has reached the defined maximum temperature threshold, the red LED has turned on, indicating that the fan actuator started (Table 4.4), as defined in the proposed architecture (section 3.2).	Satisfactory
7	Fan actuator operation – low temperatures	The minimum temperature threshold is set to 25°C (Figure 4.13). The temperature sensor was positioned far from the heat source to induce the temperature decrease until the measured temperature was lower than 25°C.	The new temperature was displayed correctly in the web application dashboard (Figure 4.11). When the measured temperature has reached the defined minimum temperature threshold, the red LED has turned off, indicating that the fan actuator has been stopped (Table 4.4), as defined in the proposed architecture (section 3.2).	Satisfactory
8	Irrigation actuator – dry soil	The minimum moisture threshold is set to 45. The moisture sensor was positioned far from the humidity source to induce the humidity decrease (Figure 4.13).	The new moisture value was displayed correctly in the web application dashboard (Figure 4.11). When the measured moisture has reached the defined minimum moisture threshold, the blue LED has turned on, indicating that the irrigation actuator has been started	Satisfactory

			(Table 4.4), as defined in the proposed architecture (section 3.2).	
9	Irrigation actuator – wet soil	The maximum moisture threshold has is set to 50. The humidity sensor was positioned near a humidity source to induce the humidity increase (Figure 4.13).	The new moisture value was displayed correctly in the web application dashboard (Figure 4.11). When the measured moisture has reached the defined maximum moisture threshold, the blue LED has turned off, indicating that the irrigation actuator has been stopped (Table 4.4), as defined in the proposed architecture (section 3.2).	Satisfactory
10	Update temperature threshold	The maximum temperature threshold has been changed from 30°C to 28°C. The temperature sensor was positioned near a heat source to induce the temperature increase (Figure 4.13).	The updated temperature threshold was displayed on the web application dashboard (Figure 4.11). When the measured temperature has reached the new maximum temperature threshold the red LED has turned on, indicating that the fan actuator has been started (Table 4.4), as defined in the proposed architecture (section 3.2).	Satisfactory
12	Update moisture threshold	The minimum moisture threshold has been changed from 45 to 35. The moisture sensor was positioned far from a humidity source to induce the humidity decrease (Figure 4.13).	The updated moisture threshold was displayed correctly on the web application dashboard (Figure 4.11). When the measured moisture has reached the updated minimum moisture threshold the blue LED has turned on, indicating that the irrigation actuator has been started (Table 4.4), as defined in the proposed architecture (section 3.2).	Satisfactory
11	Manual interaction validation	In the actuators control section of the web application console (Figure 4.12), the user clicked on the button to start the temperature actuator.	The new status of the fan actuator is displayed in the web application (Figure 4.12). The red LED has turned on, indicating that the fan actuator has been started (Table 4.4).	Satisfactory

4.3.Security Risks Assessment

This section presents a security risk assessment performed to validate and mitigate security risks in the prototype. Mapping and analyzing security risks is a fundamental activity in the risk management process because this activity allows the identification and mitigation of vulnerabilities [215]. According to [215], no standard defines which information have to be collected during a risk assessment, nevertheless the information considered during the risk assessment could include network and infrastructure architecture, network details (e.g., protocols, ports, supported services, etc.), publicly available information, databases, and other data repositories, etc. Moreover, the information considered during the risk assessment can influence the risk management process and, therefore, should be defined according to the complexity of the environment [216]. Therefore, [216] presents a risk management process based on 6 steps, as shown in Figure 4.15.

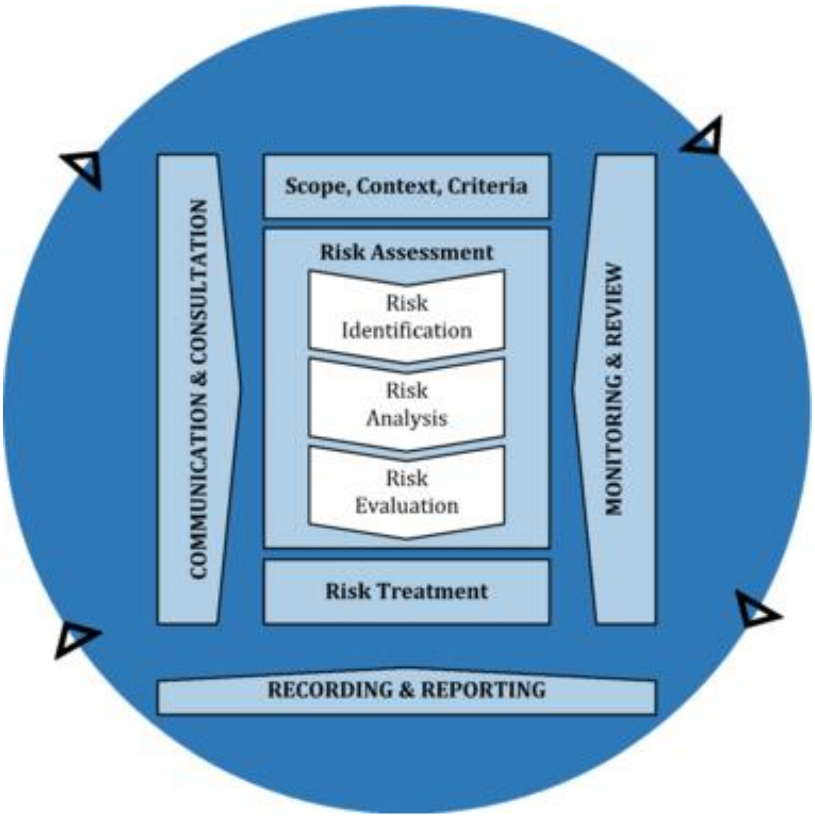


Figure 4.15 – Risk management process [216].

As shown in Figure 4.15, according to [216], the process for managing risks has 6 steps:

- Communication and consultation: assist stakeholders in understanding the risks.
- Scope, context, and criteria: define the scope of the process and understanding the context.
- Risk assessment: process that includes identification, analysis, and evaluation of risks.
- Risk treatment: select and implement options for addressing risks.

- Monitoring and review: assure and improve the quality and effectiveness of processing design, implementation, and outcomes.
- Recording and reporting: document and report through appropriate mechanisms.

The security risk assessment performed in this work aims to identify security risks on the implemented prototype, thus the steps communication and consultation, monitoring and review, and recording and reporting were not applied.

4.3.1. Scope, Context, and Criteria

The scope of this security risk analysis is the proposed architecture (discussed on section 3). The context of the security risk analysis consists of the implemented prototype (section 4). The criteria adopted for this security risk analysis took into consideration the components of the implemented prototype, such as the web application, the Raspberry Pi, and the Firebase Realtime Database. For this purpose, Table 4.9 shows the components of the prototype and the correspondence of these components with the modules of the proposed architecture presented in section 3.2.

Table 4.9 – Identification of technology assets.

Component	Module
Web application	Monitoring
Web application	Operating
Web application	Notification
Web application	Visualization
Firebase Realtime Database	Data Storage
MQTT	Application Protocol
Wi-Fi	Network Protocol
NodeMCU ESP8266 with sensors (Sensor node)	Sensor nodes
NodeMCU ESP8266 with three-segment LED (Actuator)	Actuator nodes
Raspberry Pi	Coordinator

Table 4.9 shows in the “Components” column the components of the implemented prototype, as discussed in section 4.1. Likewise, Table 4.9 shows in the "Module" column to which architecture module the prototype component corresponds.

4.3.2. Risk Assessment

According to [217], a sequence of tasks can be used for the security risk assessment step. As presented in Figure 4.16, these tasks include identifying the source of threats, identifying vulnerabilities and predisposing conditions, determining the likelihood of occurrence, and determining the impact on the environment in case of occurrence.

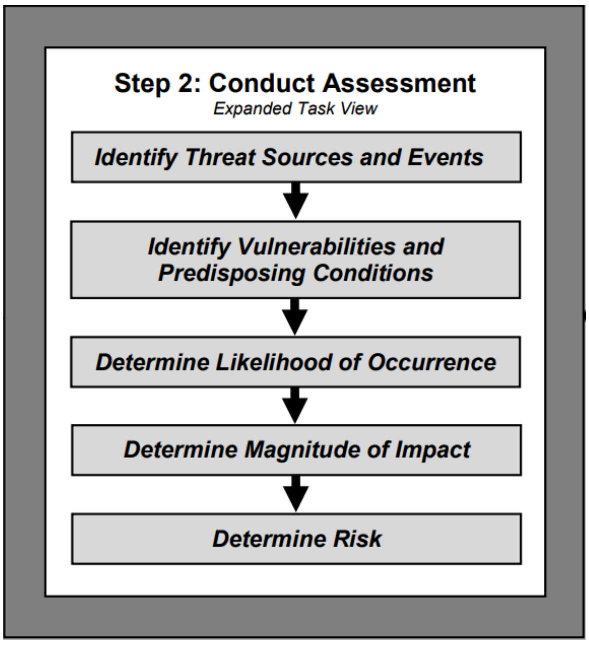
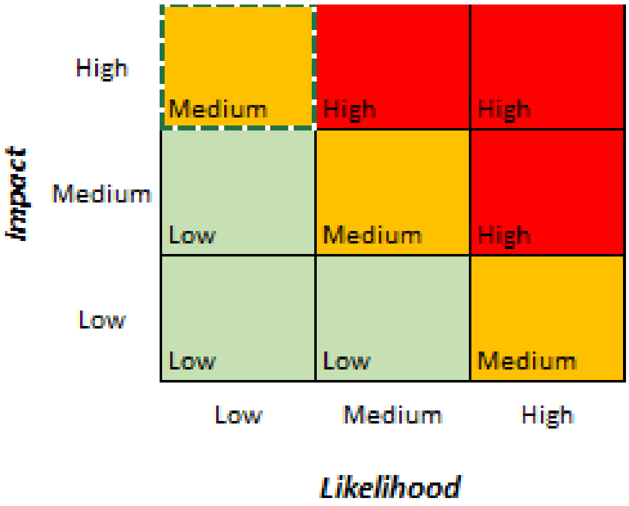

Step 2: Conduct Assessment
Expanded Task View
Identify Threat Sources and Events
Identify Vulnerabilities and Predisposing Conditions
Determine Likelihood of Occurrence
Determine Magnitude of Impact
Determine Risk

Figure 4.16 – Security risk assessment detail [217].

By knowing the impact of a given vulnerability and the likelihood of its occurrence, it is possible to classify the risk, as shown in Figure 4.17 [217], [218].



Impact			
High	Medium	High	High
Medium	Low	Medium	High
Low	Low	Low	Medium
	Low	Medium	High
	Likelihood		

Figure 4.17 - Risk classification matrix.

As presented in Figure 4.17, the risk classification is a relation between the likelihood and the impact of a given vulnerability. Thus, a vulnerability with high impact and high likelihood results in a risk classified as high, for example. Based on the risk assessment process presented in Figure 4.16, it was identified the vulnerabilities for the modules listed in Table 4.9. The criteria used to classify the vulnerabilities and their impact and likelihood are presented in Table 4.10.

Table 4.10 – Criteria used to classify the vulnerabilities and their impact and likelihood.

Criteria	Classification	Description
Vulnerability	Very high	High exposure; no response strategy.
	High	High exposure; partial response strategy.
	Medium	Moderate exposure: there is a response strategy.
	Low	Low exposure; with or without response strategies.
Impact	High	The exploitation of this vulnerability may constitute a serious anomaly in the operation, significantly compromising its overall operation.
	Medium	The exploitation of this vulnerability may constitute a localized anomaly, the impact being restricted to a group of critical processes or resources.
	Low	The exploitation of this vulnerability may constitute an isolated anomaly.
Likelihood	High	More than one annual occurrence is known.
	Medium	There is knowledge of an annual occurrence.
	Low	There is no knowledge of occurrences.

Based on Table 4.10 the impact and likelihood of each vulnerability were established. Thus, based on the impact and likelihood of each vulnerability the risk was classified, as shown in

Table 4.11. Moreover, the detailed analysis of each vulnerability is presented in appendix A.

Table 4.11 - Classification of vulnerabilities and risks.

ID	Component	Threat	Vulnerability	Likelihood	Impact	Risk classification
R.01	Web application	Exposure of sensitive data	Prototype can allow the configuration of non-secure passwords for accessing to the Web Application	Low	Low	Low
R.02	Web application	Password stealing	Users can store passwords in non-secure places	Medium	Low	Low

R.03	MQTT	Inappropriately access to data through the MQTT broker	Use of default settings in MQTT	Medium	Medium	Medium
R.04	Raspberry Pi	Unauthorized access to the Raspberry Pi via internet	Raspberry Pi is connected to the Internet	Medium	Medium	Medium
R.05	Sensor Node, Actuators	Unauthorized access to the Sensor Node via internet	Users can specify a public IP address in the configuration mode and connect the sensor node directly to the Internet	High	Medium	High
R.06	Wi-Fi	Exposure of sensitive data	Wi-Fi network can be configured without security settings or with inadequate security settings	Medium	Medium	Medium
R.07	Raspberry Pi	Compromising Wi-Fi network credentials	Wi-Fi network can be configured without security settings or with inadequate security settings	Medium	Medium	Medium
R.08	Wi-Fi	Compromising Wi-Fi network credentials	Wi-Fi network credentials can be stored at insecure locations	Medium	Low	Low
R.09	Raspberry Pi	Compromising Wi-Fi network credentials	Wi-Fi network credentials can be stored as open text within the application code	Medium	Medium	Medium
R.10	Sensor Nodes, Actuators	Unauthorized access to the Wi-Fi network through the sensor nodes	The sensor node and the actuator are implemented with NodeMCU ESP8266. NodeMCU ESP8266 offers limited support for safer security settings on Wi-Fi networks.	Low	Medium	Low
R.11	Raspberry Pi	Unauthorized access to the Raspberry Pi via Wi-Fi network	Raspberry Pi is connected to the Wi-Fi network	Medium	Medium	Medium

R.12	Raspberry Pi	MQTT broker overload	The response time of the MQTT broker can increase in the case of a large number of connected nodes	Low	Low	Low
R.13	Raspberry Pi	IoT solution impaired due to a power outage	Raspberry Pi does not have an internal power supply	Low	Medium	Low
R.14	Raspberry Pi	Interception of messages	Messages exchanged between system components can be intercepted	Low	Low	Low
R.15	MQTT	Unauthorized access to the MQTT Broker	Information within the MQTT broker cannot be encrypted	Medium	Medium	Medium
R.16	MQTT	Unauthorized access to the MQTT credentials	MQTT access credentials can be stored at insecure locations	Medium	Medium	Medium
R.17	MQTT	Compromising MQTT network credentials	MQTT access credentials can be stored in open text within the application code	Medium	Medium	Medium
R.18	MQTT	Unauthorized access to the MQTT broker settings	MQTT can be configured without access credentials or with unsafe access credentials	Low	Medium	Low
R.19	Raspberry Pi	Use of third-party software	Use of third-party Python libraries in the prototype may add unknown vulnerabilities	Low	Medium	Low
R.20	Sensor Nodes, Actuators	Use of third-party software	Use of third-party Arduino libraries in the prototype may add unknown vulnerabilities	Low	Medium	Low
R.21	Web application	Use of third-party software	Use of third-party libraries in the Web Application code may add unknown vulnerabilities	Low	Medium	Low
R.22	MQTT	Unauthorized subscription to topics of MQTT	By default, any client can subscribe to any topic in MQTT	High	Medium	High

R.23	MQTT	Unauthorized publishing of messages in Topics of MQTT	By default, any client can publish messages on topics of MQTT.	High	Medium	High
R.24	Raspberry Pi	Unauthorized physical access to the Raspberry Pi	Raspberry Pi installation location may allow improper access to the Raspberry Pi	Medium	Low	Low
R.25	Raspberry Pi	Unauthorized remote access to the Raspberry Pi	Protocols for remote access (e.g., VNC, SSH, etc.) are enabled by default in Raspberry Pi	Low	Low	Low
R.26	MQTT	Unauthorized access to the MQTT	Unauthorized access to the MQTT via the Raspberry Pi console	Medium	Medium	Medium
R.27	Raspberry Pi	Increased attack surface due to unnecessary services running on the server	By default, the SO of the Raspberry Pi executes services that are not necessary for the operation of the prototype	Medium	Low	Low
R.28	Raspberry Pi, Sensor Nodes, Actuator	Physical damage due to weather conditions	Physical damage caused by temperature or humidity due to the installation location of the Raspberry Pi, sensor nodes and actuators	Medium	Medium	Medium
R.29	Firebase Realtime Database	Database unavailable	The Firebase Realtime Database is hosted by a cloud provider and depends on the Internet to be accessed	Medium	Medium	Medium
R.30	Firebase Realtime Database	Exposure of the database on the Internet	The Firebase Realtime Database is hosted by a cloud provider and may be accessible via Internet	Medium	Medium	Medium
R.31	Firebase Realtime Database	Data may become corrupted	Data stored in the Firebase Realtime Database may become corrupted	Low	Medium	Low

Table 4.12 summarizes the number of risks identified for the components of the prototype based on the classification of the risk.

Table 4.12 - Summary of risk classification by component of the prototype.

Component of the prototype	Risk classification		
	Low	Medium	High
Web application	R.01, R.02, R.21		
Firebase Realtime Database	R.31	R.29, R.30	
MQTT	R.18	R.03, R.15, R.16, R.17, R.26	R.22, R.23
Wi-Fi	R.08	R.06	
NodeMCU ESP8266 with sensors (Sensor Node)	R.10, R.20	R.28	R.05
NodeMCU ESP8266 with three-segment LED (Actuator)	R.10, R.20	R.28	R.05
Raspberry Pi	R.12, R.13, R.14, R.19, R.24, R.25, R.27	R.04, R.07, R.09, R.11, R.28,	

As shown in Table 4.12, Raspberry Pi is the component with the highest number of identified risks. It was identified 12 risks for the Raspberry Pi, whereas 7 risks were classified as low and 5 risks were classified as medium. This large amount of risks identified for the Raspberry Pi can be justified because the Raspberry Pi hosts other components of the prototype, such as the MQTT. For the MQTT it was identified 8 risks, whereas 1 are classified as low, 5 are classified as medium and 2 are classified as high. Likewise, as presented in Table 4.12, it was identified 4 risks for sensor node and actuators, whereas 2 are classified as low, 1 is classified as medium, and 1 is classified as high. The risks identified for sensor node and actuator is the same because such risks relate to the NodeMCU ESP8266 that was used for both sensor nodes and actuators in the implemented prototype.

4.3.3. Risk Treatment

Based on the vulnerabilities identified in section 4.3.2, mechanisms for controlling and mitigating risks have been defined. These mechanisms aim to minimize the likelihood, or the impact caused by the exploitation of the vulnerabilities identified in the prototype. Table 4.13 presents the mechanisms for control and mitigation of security risks of the prototype.

Table 4.13 - Mechanisms for control and mitigation of security risks.

ID	Component	Control and Mitigation of Risk
R.01	Web application	Create policy for use of secure password in the application; Implement a mechanism that enforces the use of strong passwords;

R.02	Web application	Use a secure password repository; Define a policy for periodic password changing;
R.03	MQTT	Do not use the default settings of MQTT. Implement network security tools and mechanisms to prevent unauthorized access;
R.04	Raspberry Pi	Implement a network security system and remote access prevention mechanisms on the network; Require secure user and password for remote server access; Implement access control and logs to enable traceability;
R.05	Sensor Node, Actuators	Ensure that the monitoring modules are connected to the remote platform through the coordinator node;
R.06	Wi-Fi	Create network configuration policies and require private network access credentials; Monitor private network security settings;
R.07	Raspberry Pi	Create network configuration policies and require private network access credentials; Monitor private network security settings;
R.08	Wi-Fi	Create policies for storing network access credentials; Use a secure password repository; Monitor the security settings of the local network;
R.09	Raspberry Pi	Create policy to not store safety information in the application code; Create policies for storing network access credentials; Monitor the security settings of the local network;
R.10	Sensor Nodes, Actuators	Use libraries in the NodeMCU that allow the use of secure Wi-Fi network settings; Set a password for Wi-Fi network access;
R.11	Raspberry Pi	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Enable logs in Raspberry Pi;
R.12	Raspberry Pi	Monitor Raspberry Pi resources; Implementing authentication mechanisms for the communication between sensor nodes and the MQTT server;
R.13	Raspberry Pi	Implement a backup for power supply;
R.14	Raspberry Pi	Implement peer-to-peer SSL encryption; Monitor private network security settings;
R.15	MQTT	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Require use of secure credentials for access to MQTT; Implement logs and monitor access to the MQTT server;
R.16	MQTT	Use a secure password repository; Define a policy for periodic password changing;
R.17	MQTT	Create policy to not store safety information in the application code; Create policies for storing network access credentials; Monitor the security settings of the local network;
R.18	MQTT	Implement authentication mechanisms for the communication between sensor nodes and the MQTT server; Enforce use of secure credentials for accessing the MQTT broker; Monitor the security settings of the local network;

R.19	Raspberry Pi	Monitor the risks and implement the security updates provided by the manufacturer whenever applicable;
R.20	Sensor Nodes, Actuators	Monitor the risks and implement the security updates provided by the manufacturer whenever applicable;
R.21	Web application	Monitor the risks and implement the security updates provided by the manufacturer whenever applicable;
R.22	MQTT	Implementing authentication mechanisms for the communication between sensor nodes and the MQTT server; Enforce use of secure credentials for accessing the MQTT broker;
R.23	MQTT	Implementing authentication mechanisms for the communication between sensor nodes and the MQTT server; Enforce use of secure credentials for accessing the MQTT broker;
R.24	Raspberry Pi	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Enable logs in Raspberry Pi;
R.25	Raspberry Pi	Disable VNC protocol on the Raspberry Pi
R.26	MQTT	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Enforce use of secure credentials for accessing the MQTT broker; Enable logs in Raspberry Pi;
R.27	Raspberry Pi	Disable unnecessary services in the system; Implement security updates for the remaining services; Monitor the services running on the server;
R.28	Raspberry Pi, Sensor Nodes, Actuator	Install Raspberry Pi in a suitable location; Monitor server performance and temperature indicators;
R.29	Firebase Realtime Database	Implement backup for internet access; Implement mechanisms for off-line system operation;
R.30	Firebase Realtime Database	Creation of a policy to access the database; Implement secure credentials to access the database;
R.31	Firebase Realtime Database	Creation of a policy to access the database; Implement secure credentials to access the database; Monitor the database access;

4.4.Summary

In section 4, a prototype was implemented to validate the proposed architecture. This section was divided into 3 parts: implementation, testing, and assessment of security risks of the prototype. The first part of the section comprised the components of the prototype, such as hardware devices, the database, and developed applications. The operation and communication flow of each component were also discussed. Finally, the web application was presented. The web application allows users to interact with the IoT solution and visualize the data collected by the sensor nodes. In the second part of section 4, the test plan and the results of the performed tests were presented. The test plan considered the characteristics of the implemented prototype and aimed to validate the operation of the IoT solution. It was possible to observe through the results of the performed tests that the prototype worked properly, successfully validating the architecture. Finally, the third part of section 4 comprised the assessment of the security risks of the prototype. The third part of section 4 started by presenting a brief contextualization about security risks and the methodology used to survey the security risks. Subsequently, the security risks have been identified. Finally, the third part of section 4 presented mechanisms for controlling and mitigating the security risks.

5. Conclusions and Future Work

This work presented and validated an architecture of an IoT solution that allows monitoring and mitigating problems in plantations in various agricultural scenarios (e.g., indoor, and outdoor), as well as in plantations of different sizes. The architecture proposed in this dissertation also allows the use of big data and machine learning for the processing of data collected from plantations.

A systematic review of IoT solutions for smart farming (section 2.2) showed that traditional IoT solutions consisted of decision support systems that operated in a reactive way to data collected in real time. In contrast, in more recent work, it has been observed that management systems use complementary technologies, such as big data and machine learning, to process large amounts of data. The systematic review revealed that the most common application of IoT for smart farming is the monitoring of crops. For developing the hardware components of the IoT solutions it is used embedded system platforms (e.g., Raspberry Pi, Arduino, ESP). Different communication technologies may be used simultaneously in IoT solutions for smart farming to enable communication in short distances (e.g., BLE, ZigBee, Wi-Fi) or long distances (LoRa, Sigfox, Cellular). In addition, the comparison of types of network connections used in IoT solutions for smart farming revealed that wired networks are used in indoor scenarios (e.g., greenhouse) while wireless networks are used both in indoor and outdoor scenarios (e.g., arable lands, orchards).

Based on the systematic review of IoT solutions for smart farming, this work proposed an architecture of an IoT solution for crop monitoring capable of providing real-time crop information. The proposed architecture is modulated in 4 layers (perception, transport, processing and, application), as presented in section 2, based on the 4-layer IoT solution architecture discussed on section 2.1.2. The proposed architecture includes modules for storage, visualization, and processing of data. Moreover, the architecture proposed in this work allows the use of complementary modules for processing the data collected, such as big data and machine learning. An IoT solution that implements the proposed architecture is capable of collecting data from plantations (such as soil moisture and temperature) and implements mechanisms to mitigate and correct problems identified on the crops through actuators (e.g., ventilation and irrigation systems). Data collected by the sensor nodes are transformed in information for the users. The information obtained from the collected data can be used to monitor the plantations and automatically control the operation of actuators. Moreover, the proposed architecture allows the interaction of the user with the application to change parameters of the application in real time and to manually control actuators. In the proposed architecture the communication between sensor nodes, actuators ant the remote platform is intermediated by a coordinator node. The coordinator node enables the flexibility of the IoT solution that implements the proposed architecture, by allowing the inclusion and removal of sensor nodes and actuators. As identified on section 2.2, IoT solutions for smart farming can implement different communication technologies within a WSN. These communication technologies enable, for example, communication in short distances and with low energy consumption. For this reason, in the proposed architecture (section 3), the coordinator node is responsible for receiving messages from sensor nodes through different communication technologies (e.g., BLE, Wi-Fi, ZigBee) and retransmitting the data to a remote platform via Internet. Moreover, coordinator nodes can also retransmit messages received from sensor nodes to another coordinator nodes, allowing the monitoring of extensive plantations.

To validate the architecture proposed, a prototype that uses the technologies identified in the systematic review (section 2.2) was developed and implemented. Using IoT concepts, the IoT solution implemented in the prototype uses technologies that allow the information collected by sensors to be accessible anywhere and at any time, regardless of the device accessing it (e.g., tablets, smartphones, laptops, among others). In the prototype (section 4), the sensor node and actuators are implemented with the NodeMCU ESP8266. Likewise, the coordinator node is implemented with a Raspberry Pi 3 Model B. As discussed in section 2.2, this model of Raspberry Pi has several communication interfaces embedded, such as Ethernet and Wi-Fi. Therefore, the communication between the Raspberry Pi (coordinator) and the NodeMCU (sensor node and actuator) occurs via Wi-Fi. Due to the ubiquitous utilization of Wi-Fi in the daily life, the use of Wi-Fi simplified the implementation of the prototype in the environment used for testing and, at the same time, made it possible to validate all the objectives of this work. Additionally, the Raspberry Pi was customized in the prototype to work simultaneously as “access point” and client of a Wi-Fi network. This feature enables the Raspberry Pi to use the same network interface card to provide a private Wi-Fi network to the sensor nodes and actuators while connecting to the remote platform via internet. The private network provided by the Raspberry Pi enabled that sensor nodes and actuators could communicate with the coordinator through a private and isolated network. Moreover, Raspberry Pi does not need to use a wired network to communicate with the remote platform. Thus, the IoT solution implemented with the prototype can be used both in indoor and outdoor agriculture, as discussed in section 2.2.2. According to the proposed architecture, the data collected by the sensor nodes are transmitted by the coordinator nodes to a remote platform. The data are stored in a database and made available for processing and visualization. Additionally, it is possible to use big data and machine learning to process the stored data. In the implemented prototype the data are stored in the Firebase Realtime Database, which is a real-time cloud-based database. Firebase Realtime Database allows data to be accessible anywhere and at any time and enables the use of machine learning and big data modules. In addition, since it is cloud-based, Firebase Realtime Database enables the scalability of the IoT solution implemented in the prototype. For data visualization, a web application that uses responsive technologies was developed. This web application allows that data stored in the database are accessible from any devices connected to the internet (e.g., smartphones, tablets, etc.). Further, it was implemented mechanisms in the web application to it possible to change monitoring parameters or to control the status of actuators in real time. Finally, it was conducted operational tests on the prototype that evidenced the operation of the IoT solution. Furthermore, an assessment of the security risks of the IoT solution implemented by the prototype was carried out. Based on the identified security risks, it was proposed mechanisms for controlling and mitigating security risks.

In summary, the main objective of this dissertation was to propose an architecture of an IoT solution capable of monitoring plantations of different extensions and types while enabling the usage of machine learning and IoT. The results discussed in section 4 showed that the proposed architecture (section 3) enables the implementation of IoT solutions for monitoring and mitigating problems in plantations. This proposed architecture enables real-time data collection from plantations of different types and extensions in different agricultural scenarios (indoor and outdoor, section 2.2.2). Moreover, this architecture allows the usage of big data and machine learning modules for processing the data collected by sensor nodes. The secondary objectives of this dissertation were addressed in section 2.2 with a systematic review of the state of the art of the IoT adoption in smart agriculture and the identification of the most commonly used technologies that enable IoT solutions for smart farming. The results of section 2.2 were published in [25].

The results obtained with the prototype in a small and controlled environment were positive and successfully validated the proposed architecture using sensors to monitor the temperature and the soil moisture. Thus, the results discussed in this dissertation motivate future research, investigating the use of sensors that allow monitoring other aspects of plantations, such as luminosity, location, pressure, etc., as well as the use of different communication technologies that enable low power wide area network, such as LoRaWAN and Sigfox (section 2.1.2). This is particularly important to evaluate the reliability of the architecture with a greater amount of data of a different type and structure. Another example of future work is the implementation of the proposed architecture in a real scenario. This is particularly important to handle challenges that are intrinsic to the real plantations (e.g. noise, whether conditions, faulty sensors, etc.). In terms of safety risk, the assessment performed in section 4.3 considered only the risks related to the implemented prototype (section 4). However, the use of different technologies for the implementation of the proposed architecture may lead to different and additional security risks from the ones identified in section 4.3. In future works the security risk assessment could be extended to consider different implementations of the proposed architecture. Finally, future work could additionally include an analysis of the impact of energy consumption in IoT devices in different scenarios of agriculture (outdoor or indoor) be conducted.

Bibliographic References

[1] United Nations, Department of Economic and Social Affairs, and Population Division, “World Population Prospects 2019: Highlights,” 2019. [Online]. Available: https://population.un.org/wpp/Publications/Files/WPP2019_Highlights.pdf.

[2] D. Satterthwaite, “The implications of population growth and urbanization for climate change,” *Environment and Urbanization*, vol. 21, no. 2, pp. 545–567, Oct. 2009, doi: 10.1177/0956247809344361.

[3] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, “Big Data in Smart Farming – A review,” *Agricultural Systems*, vol. 153, no. 35, pp. 69–80, May 2017, doi: 10.1016/j.agsy.2017.01.023.

[4] D. Pivoto, P. D. Waquil, E. Talamini, C. P. S. Finocchio, V. F. D. Corte, and G. de V. Mores, “Scientific development of smart farming technologies and their application in Brazil,” *Information Processing in Agriculture*, vol. 5, no. 1, pp. 21–32, Mar. 2018, doi: 10.1016/j.inpa.2017.12.002.

[5] L. Touseau and N. Sommer, “Contribution of the Web of Things and of the Opportunistic Computing to the Smart Agriculture: A Practical Experiment,” *Future Internet*, vol. 11, no. 2, p. 33, Feb. 2019, doi: 10.3390/fi11020033.

[6] D. Ko, Y. Kwak, and S. Song, “Real Time Traceability and Monitoring System for Agricultural Products Based on Wireless Sensor Network,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 6, p. 832510, Jun. 2014, doi: 10.1155/2014/832510.

[7] E. G. Moraes, C. E. F. Gama, and W. W. Prudente, “Implantação da IoT na agricultura de precisão para eficiência hídrica na irrigação,” in *17° Congresso Nacional de Iniciação Científica*, 2017, no. 1, pp. 1–11, [Online]. Available: http://conic-semesp.org.br/anais/files/2017/trabalho-1000024621.pdf.

[8] V. V. hari Ram, H. Vishal, S. Dhanalakshmi, and P. M. Vidya, “Regulation of water in agriculture field using Internet Of Things,” in *2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR)*, Jul. 2015, pp. 112–115, doi:

10.1109/TIAR.2015.7358541.

[9] I. Potamitis, P. Eliopoulos, and I. Rigakis, “Automated Remote Insect Surveillance at a Global Scale and the Internet of Things,” *Robotics*, vol. 6, no. 3, p. 19, Aug. 2017, doi: 10.3390/robotics6030019.

[10] Z. Zhai, J.-F. Martínez Ortega, N. Lucas Martínez, and J. Rodríguez-Molina, “A Mission Planning Approach for Precision Farming Systems Based on Multi-Objective Optimization,” *Sensors*, vol. 18, no. 6, p. 1795, Jun. 2018, doi: 10.3390/s18061795.

[11] S. Kim, M. Lee, and C. Shin, “IoT-Based Strawberry Disease Prediction System for Smart Farming,” *Sensors*, vol. 18, no. 11, p. 4051, Nov. 2018, doi: 10.3390/s18114051.

[12] A. Mateo-Aroca, G. García-Mateos, A. Ruiz-Canales, J. M. Molina-García-Pardo, and J. M. Molina-Martínez, “Remote Image Capture System to Improve Aerial Supervision for Precision Irrigation in Agriculture,” *Water*, vol. 11, no. 2, p. 255, Feb. 2019, doi: 10.3390/w11020255.

[13] S. Li *et al.*, “Potential of UAV-Based Active Sensing for Monitoring Rice Leaf Nitrogen Status,” *Frontiers in Plant Science*, vol. 9, no. December, pp. 1–14, Dec. 2018, doi: 10.3389/fpls.2018.01834.

[14] J. Xue, Y. Fan, B. Su, and S. Fuentes, “Assessment of canopy vigor information from kiwifruit plants based on a digital surface model from unmanned aerial vehicle imagery,” *International Journal of Agricultural and Biological Engineering*, vol. 12, no. 1, pp. 165–171, 2019, doi: 10.25165/j.ijabe.20191201.4634.

[15] E. C. Leonard, “Precision Agriculture,” in *Encyclopedia of Food Grains*, vol. 4–4, no. June, Elsevier, 2016, pp. 162–167.

[16] M. A. Seixas and E. Contini, “Internet das coisas (IoT): inovação para o agronegócio,” Brasília, 2017. Accessed: Nov. 14, 2018. [Online]. Available: https://www.alice.cnptia.embrapa.br/bitstream/doc/1094005/1/Internetdascoisas1.pdf.

[17] Joint Research Centre (JRC) of the European Commission, P. J. Zarco-Tejada, N. Hubbard, and P. Loudjani, “OLD_Precision Agriculture: An Opportunity for EU-Farmers – Potential Support with the CAP 2014-2020,” European Union. [Online].

Available: http://www.europarl.europa.eu/RegData/etudes/note/join/2014/529049/IPOL-AGRI_NT(2014)529049_EN.pdf.

[18] BNDES, "Estudo 'Internet das Coisas: um plano de ação para o Brasil,'" *Plano Nacional de IoT*, 2017. https://www.bndes.gov.br/wps/portal/site/home/conhecimento/pesquisaedados/estudos/estudo-internet-das-coisas-iot/estudo-internet-das-coisas-um-plano-de-acao-para-o-brasil (accessed Nov. 22, 2018).

[19] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[20] A. Villa-Henriksen, G. T. C. Edwards, L. A. Pesonen, O. Green, and C. A. G. Sørensen, "Internet of Things in arable farming: Implementation, applications, challenges and potential," *Biosystems Engineering*, vol. 191, pp. 60–84, Mar. 2020, doi: 10.1016/j.biosystemseng.2019.12.013.

[21] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 4, pp. 395–420, Jun. 2017, doi: 10.3233/AIS-170440.

[22] J. M. Talavera *et al.*, "Review of IoT applications in agro-industrial and environmental fields," *Computers and Electronics in Agriculture*, vol. 142, no. 118, pp. 283–297, Nov. 2017, doi: 10.1016/j.compag.2017.09.015.

[23] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of Things in agriculture, recent advances and future challenges," *Biosystems Engineering*, vol. 164, pp. 31–48, Dec. 2017, doi: 10.1016/j.biosystemseng.2017.09.007.

[24] C. N. Verdouw, J. Wolfert, and B. Tekinerdogan, "Internet of Things in agriculture," *CAB Reviews: Perspectives in Agriculture, Veterinary Science, Nutrition and Natural Resources*, vol. 11, no. 035, Dec. 2016, doi: 10.1079/PAVSNNR201611035.

[25] E. Navarro, N. Costa, and A. Pereira, "A Systematic Review of IoT Solutions for Smart Farming," *Sensors*, vol. 20, no. 15, p. 4231, Jul. 2020, doi: 10.3390/s20154231.

[26] H. J. S. Finch, A. M. (Alison M. . Samuel, Lane G. P. F., and A. J. L. Wiseman, *Lockhart and Wiseman's crop husbandry including grassland*, 9th ed. 2002.

[27] Beecham, "Towards Smart Farming: Agriculture Embracing the IoT Vision," 2014. Accessed: Nov. 16, 2018. [Online]. Available: http://www.beechamresearch.com/files/BRL Smart Farming Executive Summary.pdf.

[28] M. J. O'Grady and G. M. P. O'Hare, "Modelling the smart farm," *Information Processing in Agriculture*, vol. 4, no. 3. Elsevier, pp. 179–187, Sep. 01, 2017, doi: 10.1016/j.inpa.2017.05.001.

[29] M. Bhagat, D. Kumar, and D. Kumar, "Role of Internet of Things (IoT) in Smart Farming: A Brief Survey," no. February, pp. 141–145, 2019, doi: 10.1109/devic.2019.8783800.

[30] M. Stočes, J. Vaněk, J. Masner, and J. Pavlík, "Internet of Things (IoT) in Agriculture - Selected Aspects," *Agris on-line Papers in Economics and Informatics*, vol. VIII, no. 1, pp. 83–88, Mar. 2016, doi: 10.7160/aol.2016.080108.

[31] Ji-chun Zhao, Jun-feng Zhang, Yu Feng, and Jian-xin Guo, "The study and application of the IOT technology in agriculture," in *2010 3rd International Conference on Computer Science and Information Technology*, Jul. 2010, vol. 2, pp. 462–465, doi: 10.1109/ICCSIT.2010.5565120.

[32] C. Verdouw, H. Sundmaeker, B. Tekinerdogan, D. Conzon, and T. Montanaro, "Architecture framework of IoT-based food and farm systems: A multiple case study," *Computers and Electronics in Agriculture*, vol. 165, p. 104939, Oct. 2019, doi: 10.1016/j.compag.2019.104939.

[33] J. Chen and A. Yang, "Intelligent Agriculture and Its Key Technologies Based on Internet of Things Architecture," *IEEE Access*, vol. 7, pp. 77134–77141, 2019, doi: 10.1109/ACCESS.2019.2921391.

[34] Telecommunication Standardization Sector of ITU, "Recommendation ITU-T Y.2060: Overview of the Internet of things," Geneva, Jun. 2012. [Online]. Available: http://handle.itu.int/11.1002/1000/11559.

[35] M. A. Uddin, A. Mansour, D. Le Jeune, M. Ayaz, and el-H. M. Aggoune, "UAV-Assisted Dynamic Clustering of Wireless Sensor Networks for Crop Health Monitoring," *Sensors*, vol. 18, no. 2, p. 555, Feb. 2018, doi: 10.3390/s18020555.

[36] N. Liu, W. Cao, Y. Zhu, J. Zhang, F. Pang, and J. Ni, "Node Deployment with k-Connectivity in Sensor Networks for Crop Information Full Coverage Monitoring," *Sensors*, vol. 16, no. 12, p. 2096, Dec. 2016, doi: 10.3390/s16122096.

[37] A. Thorat, S. Kumari, and N. D. Valakunde, "An IoT based smart solution for leaf disease detection," in *2017 International Conference on Big Data, IoT and Data Science (BID)*, Dec. 2017, vol. 2018-Janua, pp. 193–198, doi: 10.1109/BID.2017.8336597.

[38] Y. Rivas-Sánchez, M. Moreno-Pérez, and J. Roldán-Cañas, "Environment Control with Low-Cost Microcontrollers and Microprocessors: Application for Green Walls," *Sustainability*, vol. 11, no. 3, p. 782, Feb. 2019, doi: 10.3390/su11030782.

[39] F. Karim, F. Karim, and A. Frihida, "Monitoring system using web of things in precision agriculture," *Procedia Computer Science*, vol. 110, pp. 402–409, 2017, doi: 10.1016/j.procs.2017.06.083.

[40] S. Navulur, A. S. C. S. Sastry, and M. N. Giri Prasad, "Agricultural Management through Wireless Sensors and Internet of Things," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, p. 3492, Dec. 2017, doi: 10.11591/ijece.v7i6.pp3492-3499.

[41] R. S. Alonso, I. Sittón-Candanedo, Ó. García, J. Prieto, and S. Rodríguez-González, "An intelligent Edge-IoT platform for monitoring livestock and crops in a dairy farming scenario," *Ad Hoc Networks*, vol. 98, p. 102047, Mar. 2020, doi: 10.1016/j.adhoc.2019.102047.

[42] FIWARE, "The Open Source platform for our smart digital future - FIWARE," 2020. https://www.fiware.org/ (accessed Apr. 25, 2020).

[43] P. Jayaraman, A. Yavari, D. Georgakopoulos, A. Morshed, and A. Zaslavsky, "Internet of Things Platform for Smart Farming: Experiences and Lessons Learnt," *Sensors*, vol. 16, no. 11, p. 1884, Nov. 2016, doi: 10.3390/s16111884.

[44] A. L. Bustamante, M. Patricio, and J. Molina, “Thinger.io: An Open Source Platform for Deploying Data Fusion Applications in IoT Environments,” *Sensors*, vol. 19, no. 5, p. 1044, Mar. 2019, doi: 10.3390/s19051044.

[45] L. M. Fernández-Ahumada, J. Ramírez-Faz, M. Torres-Romero, and R. López-Luque, “Proposal for the Design of Monitoring and Operating Irrigation Networks Based on IoT, Cloud Computing and Free Hardware Technologies,” *Sensors*, vol. 19, no. 10, p. 2318, May 2019, doi: 10.3390/s19102318.

[46] 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, Mar. 2019, doi: 10.1016/j.icte.2017.12.005.

[47] R. Sanchez-Iborra and M.-D. Cano, “State of the Art in LP-WAN Solutions for Industrial IoT Services,” *Sensors*, vol. 16, no. 5, p. 708, May 2016, doi: 10.3390/s16050708.

[48] L. L. Peterson and B. S. Davie, *Computer Networks, Fifth Edition: A Systems Approach*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[49] S. Trilles, A. González-Pérez, and J. Huerta, “A Comprehensive IoT Node Proposal Using Open Hardware. A Smart Farming Use Case to Monitor Vineyards,” *Electronics*, vol. 7, no. 12, p. 419, Dec. 2018, doi: 10.3390/electronics7120419.

[50] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP),” RFC Editor, Jun. 2014. doi: 10.17487/RFC7252.

[51] MQTT, “FAQ - Frequently Asked Questions: MQTT,” *MQTT*. http://mqtt.org/faq (accessed Feb. 11, 2020).

[52] J. Backman *et al.*, “Cropinfra research data collection platform for ISO 11783 compatible and retrofit farm equipment,” *Computers and Electronics in Agriculture*, vol. 166, no. September, p. 105008, Nov. 2019, doi: 10.1016/j.compag.2019.105008.

[53] R. Light, “MQTT man page,” *Eclipse Mosquitto*, Sep. 2018. http://mosquitto.org/man/mqtt-7.html (accessed Feb. 11, 2020).

[54] P. P. Ray, “A survey on Internet of Things architectures,” *Journal of King Saud*

University - Computer and Information Sciences, vol. 30, no. 3, pp. 291–319, Jul. 2018, doi: 10.1016/j.jksuci.2016.10.003.

[55] R. Chandra and Y. Prihastomo, “Artificial Intelligence Definition: A Review,” pp. 1–3, 2012, [Online]. Available: https://pdfs.semanticscholar.org/d959/ad041acca7570a7229e51c18a297bb7ca0b2.pdf.

[56] X. Shi *et al.*, “State-of-the-Art Internet of Things in Protected Agriculture,” *Sensors*, vol. 19, no. 8, p. 1833, Apr. 2019, doi: 10.3390/s19081833.

[57] F. Adenugba, S. Misra, R. Maskeliūnas, R. Damaševičius, and E. Kazanavičius, “Smart irrigation system for environmental sustainability in Africa: An Internet of Everything (IoE) approach,” *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5490–5503, 2019, doi: 10.3934/mbe.2019273.

[58] S. Li *et al.*, “Combining Color Indices and Textures of UAV-Based Digital Imagery for Rice LAI Estimation,” *Remote Sensing*, vol. 11, no. 15, p. 1763, Jul. 2019, doi: 10.3390/rs11151763.

[59] S. Lee, Y. Jeong, S. Son, and B. Lee, “A Self-Predictable Crop Yield Platform (SCYP) Based On Crop Diseases Using Deep Learning,” *Sustainability*, vol. 11, no. 13, p. 3637, Jul. 2019, doi: 10.3390/su11133637.

[60] Rowe, Dawkins, and Gebhardt-Henrich, “A Systematic Review of Precision Livestock Farming in the Poultry Sector: Is Technology Focussed on Improving Bird Welfare?,” *Animals*, vol. 9, no. 9, p. 614, Aug. 2019, doi: 10.3390/ani9090614.

[61] I. Bhakta, S. Phadikar, and K. Majumder, “State-of-the-art technologies in precision agriculture: a systematic review,” *Journal of the Science of Food and Agriculture*, vol. 99, no. 11, pp. 4878–4888, Aug. 2019, doi: 10.1002/jsfa.9693.

[62] G. J. Parisoto, S. O. Gil, I. Schreinert, and L. De, “Smart Farming e seu Estado da Arte: Uma Revisão Bibliométrica,” in *VI Simpósio da Ciência e do Agronegócio*, 2018, pp. 342–351, [Online]. Available: https://www.ufrgs.br/cienagro/wp-content/uploads/2019/05/Anais-do-VI-CIENAGRO-2018.pdf.

[63] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, “Preferred Reporting Items for

Systematic Reviews and Meta-Analyses: The PRISMA Statement," *Journal of Clinical Epidemiology*, vol. 62, no. 10, pp. 1006–1012, Oct. 2009, doi: 10.1016/j.jclinepi.2009.06.005.

[64] Q. Cao, Y. Miao, J. Shen, F. Yuan, S. Cheng, and Z. Cui, "Evaluating Two Crop Circle Active Canopy Sensors for In-Season Diagnosis of Winter Wheat Nitrogen Status," *Agronomy*, vol. 8, no. 10, p. 201, Sep. 2018, doi: 10.3390/agronomy8100201.

[65] T. Zhang, W. Zhou, F. Meng, and Z. Li, "Efficiency Analysis and Improvement of an Intelligent Transportation System for the Application in Greenhouse," *Electronics*, vol. 8, no. 9, p. 946, Aug. 2019, doi: 10.3390/electronics8090946.

[66] X. Zhang, J. Zhang, L. Li, Y. Zhang, and G. Yang, "Monitoring Citrus Soil Moisture and Nutrients Using an IoT Based System," *Sensors*, vol. 17, no. 3, p. 447, Feb. 2017, doi: 10.3390/s17030447.

[67] L. Han, G. Yang, H. Yang, B. Xu, Z. Li, and X. Yang, "Clustering Field-Based Maize Phenotyping of Plant-Height Growth and Canopy Spectral Dynamics Using a UAV Remote-Sensing Approach," *Frontiers in Plant Science*, vol. 9, no. November, pp. 1–18, Nov. 2018, doi: 10.3389/fpls.2018.01638.

[68] J. Ni, J. Zhang, R. Wu, F. Pang, and Y. Zhu, "Development of an Apparatus for Crop-Growth Monitoring and Diagnosis," *Sensors*, vol. 18, no. 9, p. 3129, Sep. 2018, doi: 10.3390/s18093129.

[69] D. Fisher, L. Woodruff, S. Anapalli, and S. Pinnamaneni, "Open-Source Wireless Cloud-Connected Agricultural Sensor Network," *Journal of Sensor and Actuator Networks*, vol. 7, no. 4, p. 47, Nov. 2018, doi: 10.3390/jsan7040047.

[70] F. Muzafarov and A. Eshmuradov, "Wireless sensor network based monitoring system for precision agriculture in Uzbekistan," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 17, no. 3, p. 1071, Jun. 2019, doi: 10.12928/telkomnika.v17i3.11513.

[71] R. Bhimanpallewar and M. Rama Narasingarao, "A prototype model for continuous agriculture field monitoring and assessment," *International Journal of Engineering & Technology*, vol. 7, no. 2.7, p. 179, Mar. 2018, doi: 10.14419/ijet.v7i2.7.10288.

[72] K. Haseeb, I. U. Din, A. Almogren, and N. Islam, “An energy efficient and secure IoT-based WSN framework: An application to smart agriculture,” *Sensors (Switzerland)*, vol. 20, no. 7, 2020, doi: 10.3390/s20072081.

[73] D. Thakur, Y. Kumar, and S. Vijendra, “Smart Irrigation and Intrusions Detection in Agricultural Fields Using I.o.T.,” *Procedia Computer Science*, vol. 167, no. 2019, pp. 154–162, 2020, doi: 10.1016/j.procs.2020.03.193.

[74] M. Idbella, M. Iadaresta, G. Gagliarde, A. Mennella, S. Mazzoleni, and G. Bonanomi, “Agrilogger: A new wireless sensor for monitoring agrometeorological data in areas lacking communication networks,” *Sensors (Switzerland)*, vol. 20, no. 6, 2020, doi: 10.3390/s20061589.

[75] E. Symeonaki, K. Arvanitis, and D. Piromalis, “A context-aware middleware cloud approach for integrating precision farming facilities into the IoT toward agriculture 4.0,” *Applied Sciences (Switzerland)*, vol. 10, no. 3, 2020, doi: 10.3390/app10030813.

[76] D. Taskin and S. Yazar, “A Long-range context-aware platform design for rural monitoring with IoT In precision agriculture,” *International Journal of Computers, Communications and Control*, vol. 15, no. 2, pp. 1–11, 2020, doi: 10.15837/IJCCC.2020.2.3821.

[77] Z. Liqiang, Y. Shouyi, L. Leibo, Z. Zhen, and W. Shaojun., “A Crop Monitoring System Based on Wireless Sensor Network,” *Procedia Environmental Sciences*, vol. 11, no. PART B, pp. 558–565, 2011, doi: 10.1016/j.proenv.2011.12.088.

[78] W. Zhang *et al.*, “Research on WSN Channel Fading Model and Experimental Analysis in Orchard Environment,” in *IFIP Advances in Information and Communication Technology*, vol. 369 AICT, no. PART 2, Berlin, Heidelberg: Springer, 2012, pp. 326–333.

[79] Z. Li *et al.*, “Assimilation of Two Variables Derived from Hyperspectral Data into the DSSAT-CERES Model for Grain Yield and Quality Estimation,” *Remote Sensing*, vol. 7, no. 9, pp. 12400–12418, Sep. 2015, doi: 10.3390/rs70912400.

[80] Y. Chen, J.-P. Chanet, K.-M. Hou, H. Shi, and G. de Sousa, “A Scalable Context-Aware Objective Function (SCAOF) of Routing Protocol for Agricultural Low-Power

and Lossy Networks (RPAL),” *Sensors*, vol. 15, no. 8, pp. 19507–19540, Aug. 2015, doi: 10.3390/s150819507.

[81] N. Liu, W. Cao, Y. Zhu, J. Zhang, F. Pang, and J. Ni, “The Node Deployment of Intelligent Sensor Networks Based on the Spatial Difference of Farmland Soil,” *Sensors*, vol. 15, no. 11, pp. 28314–28339, Nov. 2015, doi: 10.3390/s151128314.

[82] D. Reynolds, J. Ball, A. Bauer, R. Davey, S. Griffiths, and J. Zhou, “CropSight: a scalable and open-source information management system for distributed plant phenotyping and IoT-based crop management,” *GigaScience*, vol. 8, no. 3, pp. 1–11, Mar. 2019, doi: 10.1093/gigascience/giz009.

[83] J. Zhang, J. Hu, L. Huang, Z. Zhang, and Y. Ma, “A Portable Farmland Information Collection System with Multiple Sensors,” *Sensors*, vol. 16, no. 10, p. 1762, Oct. 2016, doi: 10.3390/s16101762.

[84] J. Ni, L. Yao, J. Zhang, W. Cao, Y. Zhu, and X. Tai, “Development of an Unmanned Aerial Vehicle-Borne Crop-Growth Monitoring System,” *Sensors*, vol. 17, no. 3, p. 502, Mar. 2017, doi: 10.3390/s17030502.

[85] C. Granell *et al.*, “Conceptual Architecture and Service-Oriented Implementation of a Regional Geoportal for Rice Monitoring,” *ISPRS International Journal of Geo-Information*, vol. 6, no. 7, p. 191, Jun. 2017, doi: 10.3390/ijgi6070191.

[86] H. Xing *et al.*, “Global sensitivity analysis of the AquaCrop model for winter wheat under different water treatments based on the extended Fourier amplitude sensitivity test,” *Journal of Integrative Agriculture*, vol. 16, no. 11, pp. 2444–2458, Nov. 2017, doi: 10.1016/S2095-3119(16)61626-X.

[87] K. Gunasekera, A. N. Borrero, F. Vasuian, and K. P. Bryceson, “Experiences in building an IoT infrastructure for agriculture education,” *Procedia Computer Science*, vol. 135, pp. 155–162, 2018, doi: 10.1016/j.procs.2018.08.161.

[88] M. S. M, S. Das, S. Heble, U. Raj, and R. Karthik, “Internet of Things based Wireless Plant Sensor for Smart Farming,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 2, p. 456, May 2018, doi: 10.11591/ijeecs.v10.i2.pp456-468.

[89] H. Im, S. Lee, M. Naqi, C. Lee, and S. Kim, “Flexible PI-Based Plant Drought Stress Sensor for Real-Time Monitoring System in Smart Farm,” *Electronics*, vol. 7, no. 7, p. 114, Jul. 2018, doi: 10.3390/electronics7070114.

[90] F. Balducci, D. Impedovo, and G. Pirlo, “Machine Learning Applications on Agricultural Datasets for Smart Farm Enhancement,” *Machines*, vol. 6, no. 3, p. 38, Sep. 2018, doi: 10.3390/machines6030038.

[91] Y. Jeong, S. Son, S. Lee, and B. Lee, “A Total Crop-Diagnosis Platform Based on Deep Learning Models in a Natural Nutrient Environment,” *Applied Sciences*, vol. 8, no. 10, p. 1992, Oct. 2018, doi: 10.3390/app8101992.

[92] Y. Liu, K. Akram Hassan, M. Karlsson, Z. Pang, and S. Gong, “A Data-Centric Internet of Things Framework Based on Azure Cloud,” *IEEE Access*, vol. 7, pp. 53839–53858, 2019, doi: 10.1109/ACCESS.2019.2913224.

[93] C. Cambra Baseca, S. Sendra, J. Lloret, and J. Tomas, “A Smart Decision System for Digital Farming,” *Agronomy*, vol. 9, no. 5, p. 216, Apr. 2019, doi: 10.3390/agronomy9050216.

[94] R. Murugesan *et al.*, “Artificial Intelligence and Agriculture 5. 0,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 1870–1877, Jul. 2019, doi: 10.35940/ijrte.B1510.078219.

[95] X. B. Jin, X. H. Yu, X. Y. Wang, Y. T. Bai, T. L. Su, and J. L. Kong, “Deep learning predictor for sustainable precision agriculture based on internet of things system,” *Sustainability (Switzerland)*, vol. 12, no. 4, 2020, doi: 10.3390/su12041433.

[96] K. Lee, B. N. Silva, and K. Han, “Deep learning entrusted to fog nodes (DLEFN) based smart agriculture,” *Applied Sciences (Switzerland)*, vol. 10, no. 4, 2020, doi: 10.3390/app10041544.

[97] A. S. R. Murthy, Y. Sudheer, K. Mounika, K. S. Rao, and P. D. Prasad, “Cloud Technology on Agriculture using Sensors,” *Indian Journal of Science and Technology*, vol. 9, no. 17, May 2016, doi: 10.17485/ijst/2016/v9i17/93103.

[98] I. Zyrianoff *et al.*, “Architecting and deploying IoT smart applications: A performance–

oriented approach," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010084.

[99] S. H. Awan *et al.*, "BlockChain with IoT, an Emergent Routing Scheme for Smart Agriculture," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, pp. 420–429, 2020, doi: 10.14569/ijacsa.2020.0110457.

[100] R. Martínez, J. Pastor, B. Álvarez, and A. Iborra, "A Testbed to Evaluate the FIWARE-Based IoT Platform in the Domain of Precision Agriculture," *Sensors*, vol. 16, no. 11, p. 1979, Nov. 2016, doi: 10.3390/s16111979.

[101] L. Geng and T. Dong, "An agricultural monitoring system based on wireless sensor and depth learning algorithm," *International Journal of Online Engineering*, vol. 13, no. 12, pp. 127–137, Dec. 2017, doi: 10.3991/ijoe.v13i12.7885.

[102] E. A. Q. Montoya, S. F. J. Colorado, W. Y. C. Muñoz, and G. E. C. Golondrino, "Propuesta de una Arquitectura para Agricultura de Precisión Soportada en IoT," *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, vol. 24, no. 24, pp. 39–56, Dec. 2017, doi: 10.17013/risti.24.39-56.

[103] S. M. Patil and R. Sakkaravarthi, "Internet of things based smart agriculture system using predictive analytics," *Asian Journal of Pharmaceutical and Clinical Research*, vol. 10, no. 13, pp. 148–152, Apr. 2017, doi: 10.22159/ajpcr.2017.v10s1.19601.

[104] G. Guandong, J. Yuchen, and X. Ke, "An IOT-based Multi-sensor Ecological Shared Farmland Management System," *International Journal of Online Engineering (iJOE)*, vol. 14, no. 03, p. 81, Mar. 2018, doi: 10.3991/ijoe.v14i03.8199.

[105] K. Aliev, M. Moazzam, S. Narejo, E. Pasero, and A. Pulatov, "Internet of Plants Application for Smart Agriculture," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 4, pp. 421–429, 2018, doi: 10.14569/IJACSA.2018.090458.

[106] C. Dupont, M. Vecchio, C. Pham, B. Diop, C. Dupont, and S. Koffi, "An Open IoT Platform to Promote Eco-Sustainable Innovation in Western Africa: Real Urban and Rural Testbeds," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–17, Aug. 2018, doi: 10.1155/2018/1028578.

[107] G. Junxiang and D. Haiqing, “Design of Greenhouse Surveillance System Based on Embedded Web Server Technology,” *Procedia Engineering*, vol. 23, no. 2010, pp. 374–379, 2011, doi: 10.1016/j.proeng.2011.11.2516.

[108] L. S. Hong, Z. S. Sa, and J. Yan, “Environment Factors Monitoring System Based on CAN bus,” *International Journal of Online Engineering*, vol. 12, no. 5, p. 9, May 2016, doi: 10.3991/ijoe.v12i05.5722.

[109] U. Shasi Kiran, S. Arya, and M. Rajasekaran, “Design and Implementation of Smart and Low Cost Multi-task Farming System Using Arduino,” *International Journal of Engineering & Technology*, vol. 7, no. 2.24, p. 509, Apr. 2018, doi: 10.14419/ijet.v7i2.24.12148.

[110] E. Boonchieng, O. Chieochan, and A. Saokaew, “Smart Farm: Applying the Use of NodeMCU, IOT, NETPIE and LINE API for a Lingzhi Mushroom Farm in Thailand,” *IEICE Transactions on Communications*, vol. E101.B, no. 1, pp. 16–23, 2018, doi: 10.1587/transcom.2017ITI0002.

[111] C. Cambra, S. Sendra, J. Lloret, and R. Lacuesta, “Smart System for Bicarbonate Control in Irrigation for Hydroponic Precision Farming,” *Sensors*, vol. 18, no. 5, p. 1333, Apr. 2018, doi: 10.3390/s18051333.

[112] M. S. Azimi Mahmud, S. Buyamin, M. M. Mokji, and M. S. Z. Abidin, “Internet of Things based Smart Environmental Monitoring for Mushroom Cultivation,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 3, p. 847, Jun. 2018, doi: 10.11591/ijeecs.v10.i3.pp847-852.

[113] M. Erazo-Rodas *et al.*, “Multiparametric Monitoring in Equatorian Tomato Greenhouses (I): Wireless Sensor Network Benchmarking,” *Sensors*, vol. 18, no. 8, p. 2555, Aug. 2018, doi: 10.3390/s18082555.

[114] R. Z. Barbosa and J. E. M. P. Martins, “Design of a wireless sensor network for greenhouses temperature analysis,” *Irriga*, vol. 1, no. 1, pp. 132–138, Sep. 2018, doi: 10.15809/irriga.2018v1n1p132-138.

[115] Ö. Alpay and E. Erdem, “The Control of Greenhouses Based on Fuzzy Logic Using Wireless Sensor Networks,” *International Journal of Computational Intelligence*

Systems, vol. 12, no. 1, p. 190, 2018, doi: 10.2991/ijcis.2018.125905641.

[116] M. S. Munir, I. S. Bajwa, M. A. Naeem, and B. Ramzan, “Design and Implementation of an IoT System for Smart Energy Consumption and Smart Irrigation in Tunnel Farming,” *Energies*, vol. 11, no. 12, p. 3427, Dec. 2018, doi: 10.3390/en11123427.

[117] Y. Syafarinda, F. Akhadin, Z. E. Fitri, Yogiswara, B. Widiawan, and E. Rosdiana, “The Precision Agriculture Based on Wireless Sensor Network with MQTT Protocol,” *IOP Conference Series: Earth and Environmental Science*, vol. 207, no. 1, p. 012059, Dec. 2018, doi: 10.1088/1755-1315/207/1/012059.

[118] K. Radharamana, C. Mouli, and U. Kumar, “Web Architecture for Monitoring Field using Representational State Transfer Methods,” *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 1, pp. 84–93, Feb. 2019, doi: 10.22266/ijies2019.0228.09.

[119] V. S. Kumar, I. Gogul, M. D. Raj, S. K. Pragadesh, and J. S. Sebastin, “Smart Autonomous Gardening Rover with Plant Recognition Using Neural Networks,” *Procedia Computer Science*, vol. 93, no. September, pp. 975–981, 2016, doi: 10.1016/j.procs.2016.07.289.

[120] D. Cama-Pinto, M. Damas, J. A. Holgado-Terriza, F. Gómez-Mula, and A. Cama-Pinto, “Path Loss Determination Using Linear and Cubic Regression Inside a Classic Tomato Greenhouse,” *International Journal of Environmental Research and Public Health*, vol. 16, no. 10, p. 1744, May 2019, doi: 10.3390/ijerph16101744.

[121] X. Li, Z. Ma, J. Zheng, Y. Liu, L. Zhu, and N. Zhou, “An effective edge-assisted data collection approach for critical events in the SDWSN-based agricultural internet of things,” *Electronics (Switzerland)*, vol. 9, no. 6, 2020, doi: 10.3390/electronics9060907.

[122] G. Codeluppi, A. Cilfone, L. Davoli, and G. Ferrari, “LoraFarM: A LoRaWAN-based smart farming modular IoT architecture,” *Sensors (Switzerland)*, vol. 20, no. 7, 2020, doi: 10.3390/s20072028.

[123] T. hoon Kim, V. S. Solanki, H. J. Baraiya, A. Mitra, H. Shah, and S. Roy, “A smart, sensible agriculture system using the exponential moving average model,” *Symmetry*,

vol. 12, no. 3, pp. 1–15, 2020, doi: 10.3390/sym12030457.

[124] G. Sarat Chandra and K. Srinivas Ravi, “Effective Architecture for Greenhouse Controlling and Monitoring using Wi-Fi Peer to Peer Direct Protocol,” *Indian Journal of Science and Technology*, vol. 9, no. 17, May 2016, doi: 10.17485/ijst/2016/v9i17/92975.

[125] F. Ferrández-Pastor, J. García-Chamizo, M. Nieto-Hidalgo, J. Mora-Pascual, and J. Mora-Martínez, “Developing Ubiquitous Sensor Network Platform Using Internet of Things: Application in Precision Agriculture,” *Sensors*, vol. 16, no. 7, p. 1141, Jul. 2016, doi: 10.3390/s16071141.

[126] S. Rodríguez, T. Gualotuña, and C. Grilo, “A System for the Monitoring and Predicting of Data in Precision Agriculture in a Rose Greenhouse Based on Wireless Sensor Networks,” *Procedia Computer Science*, vol. 121, pp. 306–313, 2017, doi: 10.1016/j.procs.2017.11.042.

[127] N. Boonnam, J. Pitakphongmetha, S. Kajornkasirat, T. Horanont, D. Somkiadcharoen, and J. Prapakornpilai, “Optimal Plant Growth in Smart Farm Hydroponics System using the Integration of Wireless Sensor Networks into Internet of Things,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 3, pp. 1006–1012, Jul. 2017, doi: 10.25046/aj0203127.

[128] C. R. Algarín, J. C. Cabarcas, and A. P. Llanos, “Low-Cost Fuzzy Logic Control for Greenhouse Environments with Web Monitoring,” *Electronics*, vol. 6, no. 4, p. 71, Sep. 2017, doi: 10.3390/electronics6040071.

[129] L. Li *et al.*, “Sustainable energy management of solar greenhouses using open weather data on MACQU platform,” *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 1, pp. 74–82, Jan. 2018, doi: 10.25165/j.ijabe.20181101.2713.

[130] I. S. Laktionov, O. V. Vovna, Y. O. Bashkov, A. A. Zori, and V. A. Lebediev, “Improved Computer-oriented Method for Processing of Measurement Information on Greenhouse Microclimate,” *International Journal Bioautomation*, vol. 23, no. 1, pp. 71–86, Mar. 2019, doi: 10.7546/ijba.2019.23.1.71-86.

[131] G. Villalba *et al.*, “A Networked Sensor System for the Analysis of Plot-Scale

Hydrology," *Sensors*, vol. 17, no. 3, p. 636, Mar. 2017, doi: 10.3390/s17030636.

[132] S. Kameoka *et al.*, "A Wireless Sensor Network for Growth Environment Measurement and Multi-Band Optical Sensing to Diagnose Tree Vigor," *Sensors*, vol. 17, no. 5, p. 966, Apr. 2017, doi: 10.3390/s17050966.

[133] J. Xia *et al.*, "Hyperspectral Identification and Classification of Oilseed Rape Waterlogging Stress Levels Using Parallel Computing," *IEEE Access*, vol. 6, no. October, pp. 57663–57675, 2018, doi: 10.1109/ACCESS.2018.2873689.

[134] H. Lin, K. Cai, H. Chen, and Z. Zeng, "The Construction of a Precise Agricultural Information System Based on Internet of Things," *International Journal of Online Engineering (iJOE)*, vol. 11, no. 6, p. 10, Nov. 2015, doi: 10.3991/ijoe.v11i6.4847.

[135] N. Shashi Rekha, A. Kousar Nikhath, S. Nagini, Y. Sagar, and D. Sukheja, "Sustainable and Portable Low Cost IOT Based Terrace Model to Grow True Organic Greens," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 3223–3228, Aug. 2019, doi: 10.35940/ijeat.F8826.088619.

[136] S. Kumar, S. Mishra, P. Khanna, and Pragya, "Precision Sugarcane Monitoring Using SVM Classifier," *Procedia Computer Science*, vol. 122, pp. 881–887, 2017, doi: 10.1016/j.procs.2017.11.450.

[137] J. Pérez-Expósito, T. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "VineSens: An Eco-Smart Decision-Support Viticulture System," *Sensors*, vol. 17, no. 3, p. 465, Feb. 2017, doi: 10.3390/s17030465.

[138] K. Foughali, K. Fathallah, and A. Frihida, "Using Cloud IOT for disease prevention in precision agriculture," *Procedia Computer Science*, vol. 130, pp. 575–582, 2018, doi: 10.1016/j.procs.2018.04.106.

[139] T. Gayathri Devi, A. Srinivasan, S. Sudha, and D. Narasimhan, "Web enabled paddy disease detection using Compressed Sensing," *Mathematical Biosciences and Engineering*, vol. 16, no. 6, pp. 7719–7733, 2019, doi: 10.3934/mbe.2019387.

[140] R. Kalamatianos, I. Karydis, D. Doukakis, and M. Avlonitis, "DIRT: The Dacus Image Recognition Toolkit," *Journal of Imaging*, vol. 4, no. 11, p. 129, Oct. 2018, doi:

10.3390/jimaging4110129.

[141] C. Lammie, A. Olsen, T. Carrick, and M. Rahimi Azghadi, "Low-Power and High-Speed Deep FPGA Inference Engines for Weed Classification at the Edge," *IEEE Access*, vol. 7, no. c, pp. 51171–51184, 2019, doi: 10.1109/ACCESS.2019.2911709.

[142] I. Potamitis, I. Rigakis, N.-A. Tatlas, and S. Potirakis, "In-Vivo Vibroacoustic Surveillance of Trees in the Context of the IoT," *Sensors*, vol. 19, no. 6, p. 1366, Mar. 2019, doi: 10.3390/s19061366.

[143] K. Fathallah, M. A. Abid, and N. Ben Hadj-Alouane, "Enhancing Energy Saving in Smart Farming Through Aggregation and Partition Aware IOT Routing Protocol," *Sensors*, vol. 20, no. 10, p. 2760, 2020, doi: 10.3390/s20102760.

[144] I. Mohanraj, K. Ashokumar, and J. Naren, "Field Monitoring and Automation Using IOT in Agriculture Domain," *Procedia Computer Science*, vol. 93, no. January, pp. 931–939, 2016, doi: 10.1016/j.procs.2016.07.275.

[145] A. González-Briones, J. A. Castellanos-Garzón, Y. Mezquita Martín, J. Prieto, and J. M. Corchado, "A Framework for Knowledge Discovery from Wireless Sensor Networks in Rural Environments: A Crop Irrigation Systems Case Study," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–14, Jul. 2018, doi: 10.1155/2018/6089280.

[146] C. Kamienski *et al.*, "Smart water management platform: IoT-based precision irrigation for agriculture," *Sensors (Switzerland)*, vol. 19, no. 2, p. 276, Jan. 2019, doi: 10.3390/s19020276.

[147] N. Revathi and P. Sengottuvelan, "Real-Time Irrigation Scheduling Through IoT in Paddy Fields," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 4639–4647, Aug. 2019, doi: 10.35940/ijitee.J1183.0881019.

[148] J. A. López-Morales, J. A. Martínez, and A. F. Skarmeta, "Digital transformation of agriculture through the use of an interoperable platform," *Sensors (Switzerland)*, vol. 20, no. 4, pp. 1–20, 2020, doi: 10.3390/s20041153.

[149] N. G. S. Campos, A. R. Rocha, R. Gondim, T. L. C. da Silva, and D. G. Gomes, “Smart & green: An internet-of-things framework for smart irrigation,” *Sensors (Switzerland)*, vol. 20, no. 1, pp. 1–25, 2020, doi: 10.3390/s20010190.

[150] Y. Nikoloudakis, S. Panagiotakis, T. Manios, E. Markakis, and E. Pallis, “Composting as a Service: A Real-World IoT Implementation,” *Future Internet*, vol. 10, no. 11, p. 107, Nov. 2018, doi: 10.3390/fi10110107.

[151] L. Kamelia, M. A. Ramdhani, A. Faroqi, and V. Rifadiapriyana, “Implementation of Automation System for Humidity Monitoring and Irrigation System,” *IOP Conference Series: Materials Science and Engineering*, vol. 288, no. 1, p. 012092, Jan. 2018, doi: 10.1088/1757-899X/288/1/012092.

[152] S. S. Sheikh, A. Javed, M. Anas, and F. Ahmed, “Solar Based Smart Irrigation System Using PID Controller,” *IOP Conference Series: Materials Science and Engineering*, vol. 414, no. 1, p. 012040, Sep. 2018, doi: 10.1088/1757-899X/414/1/012040.

[153] J. Jarolímek, J. Pavlík, J. Kholova, and S. Ronanki, “Data Pre-processing for Agricultural Simulations,” *Agris on-line Papers in Economics and Informatics*, vol. 11, no. 01, pp. 49–53, Mar. 2019, doi: 10.7160/aol.2019.110105.

[154] M. K. I. Abd Rahman, M. S. Zainal Abidin, S. Buyamin, and M. S. Azimi Mahmud, “Enhanced Fertigation Control System towards Higher Water Saving Irrigation,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 3, p. 859, Jun. 2018, doi: 10.11591/ijeecs.v10.i3.pp859-866.

[155] M. Muñoz, J. D. Gil, L. Roca, F. Rodríguez, and M. Berenguel, “An iot architecture for water resource management in agroindustrial environments: A case study in almería (Spain),” *Sensors (Switzerland)*, vol. 20, no. 3, 2020, doi: 10.3390/s20030596.

[156] J. M. Domínguez-Niño, J. Oliver-Manera, J. Girona, and J. Casadesús, “Differential irrigation scheduling by an automated algorithm of water balance tuned by capacitance-type soil moisture sensors,” *Agricultural Water Management*, vol. 228, no. October 2019, p. 105880, 2020, doi: 10.1016/j.agwat.2019.105880.

[157] F. Ruan, R. Gu, T. Huang, and S. Xue, “A big data placement method using NSGA-III in meteorological cloud platform,” *Eurasip Journal on Wireless Communications and*

Networking, vol. 2019, no. 1, p. 143, Dec. 2019, doi: 10.1186/s13638-019-1456-7.

[158] Y. Mao, H. Qi, P. Ping, and X. Li, "Contamination Event Detection with Multivariate Time-Series Data in Agricultural Water Monitoring," *Sensors*, vol. 17, no. 12, p. 2806, Dec. 2017, doi: 10.3390/s17122806.

[159] S. A. Karimah, A. Rakhmatsyah, and N. A. Suwastika, "Smart pot implementation using fuzzy logic," *Journal of Physics: Conference Series*, vol. 1192, no. 1, p. 012058, Mar. 2019, doi: 10.1088/1742-6596/1192/1/012058.

[160] N. Mungai Bryan, K. Fei Thang, and T. Vinesh, "An Urban Based Smart IOT Farming System," *IOP Conference Series: Earth and Environmental Science*, vol. 268, no. 1, p. 012038, Jul. 2019, doi: 10.1088/1755-1315/268/1/012038.

[161] D. R. Vincent, N. Deepa, D. Elavarasan, K. Srinivasan, S. H. Chauhdary, and C. Iwendi, "Sensors Driven AI-Based Agriculture Recommendation Model for Assessing Land Suitability," *Sensors*, vol. 19, no. 17, p. 3667, Aug. 2019, doi: 10.3390/s19173667.

[162] R. V. Aroca, A. C. Hernandes, D. V. Magalhães, M. Becker, C. M. P. Vaz, and A. G. Calbo, "Calibration of Passive UHF RFID Tags Using Neural Networks to Measure Soil Moisture," *Journal of Sensors*, vol. 2018, pp. 1–12, 2018, doi: 10.1155/2018/3436503.

[163] L. Burton, N. Dave, R. E. Fernandez, K. Jayachandran, and S. Bhansali, "Smart Gardening IoT Soil Sheets for Real-Time Nutrient Analysis," *Journal of The Electrochemical Society*, vol. 165, no. 8, pp. B3157–B3162, May 2018, doi: 10.1149/2.0201808jes.

[164] F. R. G. Cruz, A. H. Ballado, A. K. A. Alcala, A. K. S. Legaspi, E. L. Lozada, and V. L. P. Portugal, "Wireless soil moisture detection with time drift compensation," in *AIP Conference Proceedings*, 2018, vol. 2045, p. 020061, doi: 10.1063/1.5080874.

[165] N. Jain, "WSN-AI based Cloud Computing Architectures for Energy Efficient Climate Smart Agriculture with Big Data analysis," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1.2, pp. 91–97, May 2019, doi: 10.30534/ijatcse/2019/1581.22019.

[166] M. Figueroa and C. Pope, “Root System Water Consumption Pattern Identification on Time Series Data,” *Sensors*, vol. 17, no. 6, p. 1410, Jun. 2017, doi: 10.3390/s17061410.

[167] A. Zervopoulos *et al.*, “Wireless sensor network synchronization for precision agriculture applications,” *Agriculture (Switzerland)*, vol. 10, no. 3, pp. 1–20, 2020, doi: 10.3390/agriculture10030089.

[168] P. Divya Vani and K. Raghavendra Rao, “Measurement and Monitoring of Soil Moisture using Cloud IoT and Android System,” *Indian Journal of Science and Technology*, vol. 9, no. 31, Aug. 2016, doi: 10.17485/ijst/2016/v9i31/95340.

[169] S. P. Srinivasan, J. Anitha, and R. Vijayakumar, “Integration of internet of things to reduce various losses of jatropha seed supply chain,” *IOP Conference Series: Materials Science and Engineering*, vol. 211, no. 1, p. 012007, Jun. 2017, doi: 10.1088/1757-899X/211/1/012007.

[170] W. Jiang, “An Intelligent Supply Chain Information Collaboration Model Based on Internet of Things and Big Data,” *IEEE Access*, vol. 7, pp. 58324–58335, 2019, doi: 10.1109/ACCESS.2019.2913192.

[171] J. Tervonen, “Experiment of the quality control of vegetable storage based on the Internet-of-Things,” *Procedia Computer Science*, vol. 130, pp. 440–447, 2018, doi: 10.1016/j.procs.2018.04.065.

[172] J. D. Borrero, “Sistema de trazabilidad de la cadena de suministro agroalimentario para cooperativas de frutas y hortalizas basado en la tecnología Blockchain,” *CIRIEC-España, revista de economía pública, social y cooperativa*, no. 95, pp. 71–94, Mar. 2019, doi: 10.7203/CIRIEC-E.95.13123.

[173] L. Mainetti, F. Mele, L. Patrono, F. Simone, M. L. Stefanizzi, and R. Vergallo, “An RFID-Based Tracing and Tracking System for the Fresh Vegetables Supply Chain,” *International Journal of Antennas and Propagation*, vol. 2013, no. July, pp. 1–15, 2013, doi: 10.1155/2013/531364.

[174] H. Fan, “Theoretical Basis and System Establishment of China Food Safety Intelligent Supervision in the Perspective of Internet of Things,” *IEEE Access*, vol. 7, pp. 71686–

71695, 2019, doi: 10.1109/ACCESS.2019.2919582.

[175] Y. Liao and K. Xu, “Traceability System of Agricultural Product Based on Block-chain and Application in Tea Quality Safety Management,” *Journal of Physics: Conference Series*, vol. 1288, no. 1, p. 012062, Aug. 2019, doi: 10.1088/1742-6596/1288/1/012062.

[176] T. Huang, S. Yan, F. Yang, T. Pan, and J. Liu, “Building SDN-Based Agricultural Vehicular Sensor Networks Based on Extended Open vSwitch,” *Sensors*, vol. 16, no. 1, p. 108, Jan. 2016, doi: 10.3390/s16010108.

[177] N. Tsolakis, D. Bechtsis, and D. Bochtis, “AgROS: A Robot Operating System Based Emulation Tool for Agricultural Robotics,” *Agronomy*, vol. 9, no. 7, p. 403, Jul. 2019, doi: 10.3390/agronomy9070403.

[178] Z. Gao *et al.*, “Wireless Channel Propagation Characteristics and Modeling Research in Rice Field Sensor Networks,” *Sensors*, vol. 18, no. 9, p. 3116, Sep. 2018, doi: 10.3390/s18093116.

[179] X. B. Jin, N. X. Yang, X. Y. Wang, Y. T. Bai, T. L. Su, and J. L. Kong, “Hybrid deep learning predictor for smart agriculture sensing based on empirical mode decomposition and gated recurrent unit group model,” *Sensors (Switzerland)*, vol. 20, no. 5, 2020, doi: 10.3390/s20051334.

[180] R. Dhall and H. Agrawal, “An Improved Energy Efficient Duty Cycling Algorithm for IoT based Precision Agriculture,” *Procedia Computer Science*, vol. 141, pp. 135–142, 2018, doi: 10.1016/j.procs.2018.10.159.

[181] H. Jawad, R. Nordin, S. Gharghan, A. Jawad, M. Ismail, and M. Abu-AlShaeer, “Power Reduction with Sleep/Wake on Redundant Data (SWORD) in a Wireless Sensor Network for Energy-Efficient Precision Agriculture,” *Sensors*, vol. 18, no. 10, p. 3450, Oct. 2018, doi: 10.3390/s18103450.

[182] C. Li and B. Niu, “Design of smart agriculture based on big data and Internet of things,” *International Journal of Distributed Sensor Networks*, vol. 16, no. 5, 2020, doi: 10.1177/1550147720917065.

[183] N. Sabri, S. S. Mohammed, S. Fouad, A. A. Syed, F. T. Al-Dhief, and A. Raheemah, "Investigation of Empirical Wave Propagation Models in Precision Agriculture," *MATEC Web of Conferences*, vol. 150, p. 06020, Feb. 2018, doi: 10.1051/matecconf/201815006020.

[184] M. G. González-González, J. Gómez-Sanchis, J. Blasco, E. Soria-Olivas, and P. Chueca, "CitrusYield: A dashboard for mapping yield and fruit quality of citrus in precision agriculture," *Agronomy*, vol. 10, no. 1, pp. 1–13, 2020, doi: 10.3390/agronomy10010129.

[185] M. Watanabe, A. Nakamura, A. Kunii, K. Kusano, and M. Futagawa, "Fabrication of Scalable Indoor Light Energy Harvester and Study for Agricultural IoT Applications," *Journal of Physics: Conference Series*, vol. 660, no. 1, p. 012110, Dec. 2015, doi: 10.1088/1742-6596/660/1/012110.

[186] I. García-Magariño, R. Lacuesta, and J. Lloret, "ABS-SmartComAgri: An Agent-Based Simulator of Smart Communication Protocols in Wireless Sensor Networks for Debugging in Precision Agriculture," *Sensors*, vol. 18, no. 4, p. 998, Mar. 2018, doi: 10.3390/s18040998.

[187] M. K. Gayatri, J. Jayasakthi, and G. S. A. 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, doi: 10.1109/TIAR.2015.7358528.

[188] Arduino, "Getting Started with Arduino products," 2020. https://www.arduino.cc/en/Guide/HomePage (accessed Jun. 07, 2020).

[189] Raspberry Foundation, "Raspberry Pi: Products." https://www.raspberrypi.org/products/ (accessed Mar. 28, 2020).

[190] G. Castellanos, M. Deruyck, L. Martens, and W. Joseph, "System Assessment of WUSN Using NB-IoT UAV-Aided Networks in Potato Crops," *IEEE Access*, vol. 8, pp. 56823–56836, 2020, doi: 10.1109/ACCESS.2020.2982086.

[191] A. H. Ali, R. F. Chisab, and M. J. Mnati, "A smart monitoring and controlling for

agricultural pumps using LoRa IOT technology," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 286–292, Jan. 2019, doi: 10.11591/ijeecs.v13.i1.pp286-292.

[192] W. Xue-fen, Y. Yi, Z. Tao, Z. Jing-wen, and M. S. Sardar, "Design of Distributed Agricultural Service Node with Smartphone In-field Access Supporting for Smart Farming in Beijing-Tianjin-Hebei Region," *Sensors and Materials*, vol. 30, no. 10, p. 2281, Oct. 2018, doi: 10.18494/SAM.2018.1846.

[193] H. Klaina, A. Vazquez Alejos, O. Aghzout, and F. Falcone, "Narrowband Characterization of Near-Ground Radio Channel for Wireless Sensors Networks at 5G-IoT Bands," *Sensors*, vol. 18, no. 8, p. 2428, Jul. 2018, doi: 10.3390/s18082428.

[194] V. C. Gungor *et al.*, "Smart Grid Technologies: Communication Technologies and Standards," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, Nov. 2011, doi: 10.1109/TII.2011.2166794.

[195] P. Lea, "Long-Range Communication Systems and Protocols (WAN)," in *Internet of Things for Architects: Learn to Design, Implement and secure your IoT infrastructure*, 1st ed., Birmingham: Packt Publishing, 2018, p. 524.

[196] Y. Zhang and W. W. Li, "Energy Consumption Analysis of a Duty Cycle Wireless Sensor Network Model," *IEEE Access*, vol. 7, pp. 33405–33413, 2019, doi: 10.1109/ACCESS.2019.2903303.

[197] D. Xu, S. Wu, B. Zhang, and X. Qin, "Power Balance AODV Algorithm of WSN in Agriculture Monitoring," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 11, no. 4, p. 811, Dec. 2013, doi: 10.12928/telkomnika.v11i4.1205.

[198] S. Kodati and S. Jeeva, "Smart Agricultural using Internet of Things, Cloud and Big Data," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 3718–3722, Aug. 2019, doi: 10.35940/ijitee.J9671.0881019.

[199] L. Zhao, L. He, X. Jin, and W. Yu, "Design of Wireless Sensor Network Middleware for Agricultural Applications," in *IFIP Advances in Information and Communication Technology*, vol. 393 AICT, no. PART 2, Berlin, Heidelberg: Springer, 2013, pp. 270–279.

[200] A. De Mauro, M. Greco, and M. Grimaldi, “A formal definition of Big Data based on its essential features,” *Library Review*, vol. 65, no. 3, pp. 122–135, Apr. 2016, doi: 10.1108/LR-06-2015-0061.

[201] M. Adam Ibrahim Fakherldin, K. Adam, N. Akma Abu Bakar, and M. Abdul Majid, “Weather Data Analysis Using Hadoop: Applications and Challenges,” *IOP Conference Series: Materials Science and Engineering*, vol. 551, no. 1, p. 012044, Aug. 2019, doi: 10.1088/1757-899X/551/1/012044.

[202] L. Zhang, G. Gui, A. M. Khattak, M. Wang, W. Gao, and J. Jia, “Multi-Task Cascaded Convolutional Networks Based Intelligent Fruit Detection for Designing Automated Robot,” *IEEE Access*, vol. 7, pp. 56028–56038, 2019, doi: 10.1109/ACCESS.2019.2899940.

[203] NASA, “POWER, data sets from NASA research with solar and meteorological information,” 2019. https://data.nasa.gov/Earth-Science/Prediction-Of-Worldwide-Energy-Resources-POWER-/wn3p-qsan.

[204] J. Steele, “Understanding Virtual Sensors: From Sensor Fusion To Context-Aware Applications,” *Electronic Design*, Jul. 10, 2012. http://electronicdesign.com/ios/understanding-virtual-sensors-sensor-fusion-context-aware-applications (accessed May 28, 2020).

[205] Y.-R. Chien and Y.-X. Chen, “An RFID-Based Smart Nest Box: An Experimental Study of Laying Performance and Behavior of Individual Hens,” *Sensors*, vol. 18, no. 3, p. 859, Mar. 2018, doi: 10.3390/s18030859.

[206] Statista, “Forecasted market value of precision farming worldwide in 2018 and 2023,” 2018. https://www.statista.com/statistics/721921/forecasted-market-value-of-precision-farming-worldwide/ (accessed Oct. 01, 2019).

[207] NodeMcu Team, “NodeMcu Connect Things EASY,” 2018. https://www.nodemcu.com/index_en.html#fr_54745c8bd775ef4b99000011 (accessed Oct. 04, 2019).

[208] Espressif, “ESP8266 Overview - Espressif Systems,” *Espressif*, Sep. 28, 2018. https://www.espressif.com/products/hardware/esp8266ex/overview/ (accessed Sep. 28,

2019).

[209] Maxim Integrated, “DS18B20: Programmable Resolution 1-Wire Digital Thermometer.” pp. 1–20, 2019, [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf.

[210] W. Vernandhes, N. . Salahuddin, A. Kowanda, and S. P. Sari, “Smart aquaponic with monitoring and control system based on iot,” in *2017 Second International Conference on Informatics and Computing (ICIC)*, Nov. 2017, vol. 2018-Janua, no. November, pp. 1–6, doi: 10.1109/IAC.2017.8280590.

[211] Texas Instruments, “LMx93-N, LM2903-N Low-Power, Low-Offset Voltage, Dual Comparators.” pp. 1–31, 2018, [Online]. Available: http://www.ti.com/lit/ds/symlink/lm393-n.pdf.

[212] P. Hutchinson, “Raspberry Pi 4, 3 B+, Pi 3, Pi 2, B+, A+ Comparison Chart,” *Element14*, 2019. https://www.element14.com/community/docs/DOC-68090/l/raspberry-pi-4-3-b-pi-3-pi-2-b-a-comparison-chart (accessed Mar. 28, 2020).

[213] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O’Reilly Media, Inc, 2018.

[214] A. Dufetel, “Introducing Cloud Firestore: Our New Document Database for Apps,” Oct. 03, 2017. https://firebase.googleblog.com/2017/10/introducing-cloud-firestore.html (accessed Apr. 20, 2020).

[215] R. Schmittling and A. Munns, “Performing a Security Risk Assessment,” *ISACA Journal*, vol. 1, pp. 1–7, 2010, [Online]. Available: http://www.isaca.org/Journal/Past-Issues/2010/Volume-1/Pages/Performing-a-Security-Risk-Assessment1.aspx.

[216] ISO, “Risk management - Guidelines,” *ISO 31000:2018*, 2018. https://www.iso.org/obp/ui/#iso:std:iso:31000:ed-2:v1:en (accessed Apr. 25, 2020).

[217] R. S. Ross, “Guide for Conducting Risk Assessments,” *Special Publication (NIST SP) - 800-30 Rev 1*, no. September, p. 95, 2012, doi: 10.6028/NIST.SP.800-30r1.

[218] I. Sotnikov, “How to Perform IT Risk Assessment,” *Netwrix*, Jan. 10, 2019. https://blog.netwrix.com/2018/01/16/how-to-perform-it-risk-assessment (accessed Jan.

10, 2019).

Appendix A – Security Risks of the Prototype

R.01	Exposure of sensitive data	**Classification**	Low
Vulnerability	Prototype can allow the configuration of non-secure passwords for accessing to the Web Application		
Impact evaluation	The use of insecure passwords in the application may allow exposing confidential information of the system.	Vulnerability	High
		Prior notice:	Yes
		Duration:	Medium
		Impact (V+L+D)	6
			Low
Likelihood	Password setting is inherent to the user so there is a risk that the user will use insecure passwords.	Likelihood classification	Medium
Risk control	Create policy for use of secure password in the application; Implement a mechanism that enforces the use of strong passwords;	Risk control classification	Acceptable
Residual risk	Users still have the ability to define insecure passwords or to store them in inappropriate locations, potentially allowing improper access.	Residual risk classification	Medium

R.02	Password stealing	**Classification**	Low
Vulnerability	*Users can store passwords in non-secure places*		
Impact evaluation	The user can store passwords in unsecured locations, allowing unauthorized access to the web application.	Vulnerability	Medium
		Prior notice:	Yes
		Duration:	Long
		Impact (V+L+D)	6
			Low
Likelihood	The password belongs to the user and he can store it anywhere physically or digitally.	Likelihood classification	Medium
Risk control	Use a secure password repository; Define a policy for periodic password changing;	Risk control classification	Acceptable
Residual risk	The user is still the only responsible for storing the information and the risk of passwords being compromised remains despite the mitigation responses.	Residual risk classification	Medium

R.03	Inappropriately access to data through the MQTT broker	**Classification**	Medium
Vulnerability	Use of default settings in MQTT		
Impact evaluation	The use of default settings in the MQTT allows malicious users to unduly access the MQTT, potentially exposing sensitive information from the IoT solution.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	6
			Medium
Likelihood	The MQTT is a well-established protocol and therefore its default settings are known. Thus, the default settings of MQTT can be used to exploit vulnerabilities.	Likelihood Classification	Medium
Risk control	Do not use the default settings of MQTT. Implement network security tools and mechanisms to prevent unauthorized access;	Risk control classification	Acceptable

Residual risk	There is still a risk that non-standard settings are exposed and used to attack the IoT solution. Users and passwords can be stored insecurely and listening tools on the network allow the discovery of communication ports.	Residual risk classification	Medium

R.04	*Unauthorized access to the Raspberry Pi via internet*	**Classification**	Medium
Vulnerability	*Raspberry Pi is connected to the Internet*		
Impact evaluation	In case a malicious user gains access to the coordinator node via Internet, this user may have access to the settings and sensitive data of the IoT solution.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	6
			Medium
Likelihood	The network architecture and other security mechanisms reduce the probability of this vulnerability being exploited.	Likelihood Classification	Medium
Risk control	Implement a network security system and remote access prevention mechanisms on the network; Require secure user and password for remote server access; Implement access control and logs to enable traceability;	Risk control classification	Efficient
Residual risk	As long as the coordinating node is connected to the internet this vulnerability still existing.	Residual risk classification	Low

R.05	Unauthorized access to the Sensor Node via internet	**Classification**	Medium
Vulnerability	*Users can specify a public IP address in the configuration mode and connect the sensor node or actuators directly to the Internet*		
Impact evaluation	When connected to the internet, sensor nodes and actuators can be accessed in an unauthorized way and data collected by sensor nodes can be exposed.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	The network architecture and other security mechanisms reduce the probability of this vulnerability being exploited.	Likelihood Classification	High
Risk control	Ensure that the monitoring modules are connected to the remote platform through the coordinator node;	Risk control classification	Acceptable
Residual risk	As long as the sensor node and the actuator are connected to the internet this vulnerability still existing.	Residual risk classification	Medium

R.06	Exposure of sensitive data	**Classification**	Medium
Vulnerability	Wi-Fi network can be configured without security settings or with inadequate security settings		
Impact evaluation	Configuring the private network without security allows unauthorized devices to connect to the network and exposes system settings and sensitive data.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	7

			Medium
Likelihood	The security settings of the private network can be defined by the user.	Likelihood Classification	Medium
Risk control	Create network configuration policies and require private network access credentials; Monitor private network security settings;	Risk control classification	Insufficient
Residual risk	Even with the definition of security policies there is still the possibility that the user uses insecure settings in the private network	Residual risk classification	Medium

R.07	Compromising Wi-Fi network credentials	**Classification**	Medium
Vulnerability	Wi-Fi network can be configured without security settings or with inadequate security settings		
Impact evaluation	Unsecured access credentials may allow invasion of the private network and expose configurations and sensitive data of the IoT solution.	Vulnerability	Medium
		Prior notice	No
		Duration	Medium
		Impact (V+L+D)	Medium
Likelihood	The security settings of the private network can be defined by the user.	Likelihood Classification	Medium
Risk control	Create network configuration policies and require private network access credentials; Monitor private network security settings;	Risk control classification	Insufficient
Residual risk	Even with the definition of security policies there is still the possibility that the user uses insecure settings in the private network	Residual risk classification	Medium

R.08	Compromising Wi-Fi network credentials	**Classification**	Low
Vulnerability	Wi-Fi network credentials can be stored at insecure locations		
Impact evaluation	Storing private network access credentials in unsecured locations can expose access to credentials, enabling connection to the network and exposing data and system settings.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Short
		Impact (V+L+D)	Low
Likelihood	The password belongs to the user and he can store it anywhere physically or digitally.	Likelihood Classification	Medium
Risk control	Create policies for storing network access credentials; Use a secure password repository; Monitor the security settings of the local network;	Risk control classification	Acceptable
Residual risk	Even with the definition of policies there is still the possibility of the user to infringe them and store the access credentials in unsafe locations.	Residual risk classification	Medium

R.09	Compromising Wi-Fi network credentials	**Classification**	Medium
Vulnerability	Wi-Fi network credentials can be stored as open text within the application code		
Impact evaluation	Storing private network access credentials as open text in the source code enables unauthorized access to	Vulnerability	High
		Prior notice:	No

	the private network and can expose sensitive data and IoT solution settings.	Duration:	Long
		Impact (V+L+D)	Medium
Likelihood	Applications that runs on the coordinator node are developed in Python. Since Python applications are not compiled, if the network credentials are stored in the source code of the application, a malicious user can gain access to the network credentials.	Likelihood Classification	Medium
Risk control	Create policy to not store safety information in the application code. Create policies for storing network access credentials. Monitor the security settings of the local network.	Risk control classification	Efficient
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Residual risk classification	Low

R.10	Unauthorized access to the Wi-Fi network through the sensor nodes	**Classification**	Low
Vulnerability	The Sensor node and the actuator are implemented with NodeMCU ESP8266. NodeMCU ESP8266 offers limited support for safer security settings on Wi-Fi networks.		
Impact evaluation	Using basic security settings on the private network can facilitate the discovery of access credentials and allow intrusion into the private network, potentially exposing system settings and data.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	You can use libraries to implement Wi-Fi network settings with satisfactory security levels. Private network reduces the attack surface.	Likelihood Classification	Low
Risk control	Use libraries in the NodeMCU that allow the use of secure Wi-Fi network settings; Set a password for accessing the local network;	Risk control classification	Efficient
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Residual risk classification	Low

R.11	Unauthorized access to the Raspberry Pi via Wi-Fi network	**Classification**	Medium
Vulnerability	Raspberry Pi is connected to the Wi-Fi network		
Impact evaluation	If a malicious user has access to Raspberry Pi, the user can have access to the settings and sensitive data of the IoT solution.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium

Likelihood	To exploit this vulnerability a malicious user must have access to the Wi-Fi network. If the WI-FI network is invaded, then it is possible that the unauthorized user could have improper access to Raspberry Pi.	Likelihood Classification	Medium
Risk control	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Enable logs in Raspberry Pi;	Risk control classification	Efficient
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Residual risk classification	Low

R.12	MQTT broker overload	**Classification**	Low
Vulnerability	The response time of the MQTT broker can increase in the case of a large number of connected nodes		
Impact evaluation	A large number of clients connected to the MQTT broker can increase the response time to make the service unavailable. A malicious user could exploit this vulnerability to execute a DDoS attack on the system.	Vulnerability	High
		Prior notice:	Yes
		Duration:	Short
		Impact (V+L+D)	Low
Likelihood	To exploit this vulnerability the malicious user must have access to the private network and be able to connect to the MQTT server.	Likelihood Classification	Low
Risk control	Monitor Raspberry Pi resources; Implementing authentication mechanisms for the communication between sensor nodes and the MQTT server;	Risk control classification	Acceptable
Residual risk	The problem can still occur if the quantity of sensory nodes and valid actuators exceeds the capacity of the server. However, resource monitoring allows a preventive action to mitigate this scenario.	Residual risk classification	Low

R.13	IoT solution impaired due to a power outage	**Classification**	Low
Vulnerability	Raspberry Pi does not have an internal power supply		
Impact evaluation	If Raspberry Pi becomes unavailable the IoT solution will no longer work properly.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	The power outage is a possibility that may occur due to the electric power service provider.	Likelihood Classification	Low
Risk control	Implement a backup for power supply;	Risk control classification	Efficient
Residual risk	Power outage can last longer than the capacity of the backup power supply.	Residual risk classification	Low

R.14	Interception of messages	**Classification**	Low
Vulnerability	Messages exchanged between system components can be intercepted		
Impact evaluation	A malicious user can exploit this vulnerability to intercept communication between sensory nodes, actuators, and the coordinator, and gain unauthorized access to sensitive data and IoT solution settings.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Short
		Impact (V+L+D)	Low
Likelihood	For this vulnerability to be exploited a malicious user must be able to connect to the Wi-Fi network. However, security mechanisms implemented on the Wi-Fi network decrease the probability of this vulnerability occurring.	Likelihood Classification	Low
Risk control	Implement peer-to-peer SSL encryption; Monitor private network security settings;	Risk control classification	Acceptable
Residual risk	SSL encryption can contribute to increased power consumption and processing in the components of the IoT solution (e.g. sensor nodes, actuators).	Residual risk classification	Medium

R.15	Unauthorized access to the MQTT Broker	**Classification**	Medium
Vulnerability	Information within the MQTT broker cannot be encrypted		
Impact evaluation	In case of unauthorized access to the MQTT broker, confidential information of the IoT solution will be exposed. At the same time, privileged access to the MQTT broker makes it possible to drive system actuators, adding potential impacts to the business.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	To exploit this vulnerability a malicious user must be able to connect to the server physically through the internal network or the internet and access the MQTT server.	Likelihood Classification	Medium
Risk control	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Require use of secure credentials for access to MQTT; Implement logs and monitor access to the MQTT server;	Risk control classification	Efficient
Residual risk	If the credentials for the MQTT are stored in an unsafe location, the security of the MQTT may be compromised.	Residual risk classification	Low

R.16	Unauthorized access to the MQTT credentials	**Classification**	Medium
Vulnerability	MQTT access credentials can be stored at insecure locations		
Impact evaluation		Vulnerability	High
		Prior notice:	No

	A malicious user can exploit this vulnerability to access sensitive information and change configuration in the IoT solution.	Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	To exploit this vulnerability a malicious user must have access to the information stored in an insecure way.	Likelihood Classification	Medium
Risk control	Use a secure password repository; Define a policy for periodic password changing;	Risk control classification	Acceptable
Residual risk	The user is still the only responsible for storing the information and the risk of passwords being compromised remains despite the mitigation responses	Residual risk classification	Medium

R.17	Compromising MQTT network credentials	**Classification**	Medium
Vulnerability	MQTT access credentials can be stored in open text within the application code		
Impact evaluation	Storing MQTT credentials as open text in the source code enables unauthorized access to the MQTT broker and can expose sensitive data and IoT solution settings.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	Applications that runs on the coordinator node are developed in Python. Since Python applications are not compiled, if the network credentials are stored in the source code of the application, a malicious user can gain access to the MQTT credentials.	Likelihood Classification	Medium
Risk control	Create policy to not store safety information in the application code; Create policies for storing network access credentials; Monitor the security settings of the local network;	Classification	Efficient
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Classification	Low

R.18	Unauthorized access to the MQTT broker settings	**Classification**	Low
Vulnerability	MQTT can be configured without access credentials or with unsafe access credentials		
Impact evaluation	A malicious user can exploit this vulnerability to gain access to the MQTT and obtain sensitive data and change IoT solution settings.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	To exploit this vulnerability a malicious user needs to be able to discover the MQTT user and password, have access to the Wi-Fi network and the server where MQTT is installed.	Likelihood Classification	Low
Risk control	Implement authentication mechanisms for the communication between sensor nodes and the MQTT server; Enforce use of secure credentials for accessing the MQTT broker; Monitor the security settings of the local network;	Risk control classification	Acceptable
Residual risk	MQTT has no mechanisms to force the use of secure passwords. Despite the policy, the configuration of credentials is the user's responsibility and may be violated.	Residual risk classification	Medium

R.19	Use of third-party software	**Classification**	Low
Vulnerability	Use of third-party Python libraries in the prototype may add unknown vulnerabilities		
Impact evaluation	Python libraries developed by third parties and used in the application running in Raspberry Pi add non-mapping vulnerabilities to the IoT solution.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	For a malicious user to exploit a vulnerability it is necessary to know the vulnerabilities of the libraries in use and be able to access the system components	Likelihood Classification	Low
Risk control	Monitor the risks and implement the security updates provided by the manufacturer whenever applicable;	Risk control classification	Acceptable
Residual risk	New non-mapping vulnerabilities can be introduced after patches and fixes are applied.	Residual risk classification	Low

R.20	Use of third-party software	**Classification**	Low
Vulnerability	Use of third-party Arduino libraries in the prototype may add unknown vulnerabilities		
Impact evaluation	Third party libraries used in the NodeMCU application may add non-mapping vulnerabilities to the IoT solution.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	For a malicious user to exploit a vulnerability it is necessary to know the vulnerabilities of the libraries in use and be able to access the system components	Likelihood Classification	Low
Risk control	Monitor the risks and implement the security updates provided by the manufacturer whenever applicable;	Risk control classification	Acceptable
Residual risk	New non-mapping vulnerabilities can be introduced after patches and fixes are applied.	Residual risk classification	Low

R.21	Use of third-party software	**Classification**	Low
Vulnerability	Use of third-party libraries in the Web application code may add unknown vulnerabilities		
Impact evaluation	Third party libraries used in the web application may add non-mapping vulnerabilities to the IoT solution.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	For a malicious user to exploit a vulnerability it is necessary to know the vulnerabilities of the libraries in use and be able to access the system components	Likelihood Classification	Low
Risk control	Monitor the risks and implement the security updates provided by the manufacturer whenever applicable;	Risk control classification	Acceptable
Residual risk	New non-mapping vulnerabilities can be introduced after patches and fixes are applied.	Residual risk classification	Low

R.22	Unauthorized subscription to topics of MQTT	**Classification**	High
Vulnerability	By default, any client can subscribe to any topic in MQTT		
Impact evaluation	A malicious user can exploit this vulnerability to subscribe to a topic in the MQTT in an unauthorized way and access confidential information from the IoT solution.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	To exploit this vulnerability the user needs access to the Wi-Fi network.	Likelihood Classification	High
Risk control	Implementing authentication mechanisms for the communication between sensor nodes, actuators, and the MQTT broker. Enforce use of secure credentials for accessing the MQTT broker;	Risk control classification	Acceptable
Residual risk	MQTT access credentials are defined for each MQTT broker. In case the MQTT credentials are compromised, the MQTT server will be vulnerable again.	Residual risk classification	Medium

R.23	Unauthorized publishing of messages in Topics of MQTT	**Classification**	High
Vulnerability	By default, any client can publish messages on topics of MQTT		
Impact evaluation	A malicious user can exploit this vulnerability to gain access to posting messages on MQTT topics. Posting messages on specific MQTT topics allows the user to manipulate IoT solution settings.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	To exploit this vulnerability the user needs access to the Wi-Fi network.	Likelihood Classification	High
Risk control	Implementing authentication mechanisms for the communication between sensor nodes, actuators, and the MQTT broker; Enforce use of secure credentials for accessing the MQTT broker;	Risk control classification	Acceptable

Residual risk	MQTT access credentials are defined for each MQTT broker. In case the MQTT credentials are compromised, the MQTT server will be vulnerable again.	Residual risk classification	Medium

R.24	Unauthorized physical access to the Raspberry Pi	**Classification**	Low
Vulnerability	Raspberry Pi installation location may allow improper access to the Raspberry Pi		
Impact evaluation	A malicious user can exploit this vulnerability to gain physical access to Raspberry Pi and access sensitive IoT solution settings and data.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Short
		Impact (V+L+D)	Low
Likelihood	Due to the characteristics of the environment, there is the possibility of physical access to Raspberry Pi.	Likelihood Classification	Medium
Risk control	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Enable logs in Raspberry Pi;	Risk control classification	Acceptable
Residual risk	If access credentials are compromised the malicious user can get root access to Raspberry Pi. Physical access to the server allows it to be turned off, which impacts the operation of the IoT solution.	Residual risk classification	Medium

R.25	*Unauthorized remote access to the Raspberry Pi*	**Classification**	Low
Vulnerability	*Protocols for remote access (e.g., VNC, SSH, etc.) are enabled by default in Raspberry Pi*		
Impact evaluation	A malicious user can exploit this vulnerability to gain remote access to Raspberry Pi through a VNC session. In addition, data trafficked during a VNC session is not encrypted and can be accessed improperly.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Short
		Impact (V+L+D)	Low
Likelihood	To exploit this vulnerability a malicious user needs access to the Wi-Fi network where Raspberry Pi is installed.	Likelihood Classification	Low
Risk control	Deactivate unnecessary remote access services in the Raspberry Pi	Risk control classification	Efficient
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Residual risk classification	Low

R.26	Unauthorized access to the MQTT	**Classification**	Medium
Vulnerability	Unauthorized access to the MQTT via the Raspberry Pi console		
Impact evaluation	A malicious user can exploit this vulnerability to access the MQTT through the Raspberry Pi console and gain access to the configurations and sensitive data of the IoT solution and to manipulate the behavior of the actuators in the IoT solution.	Vulnerability	High
		Prior notice:	No
		Duration:	Short
		Impact (V+L+D)	Medium
Likelihood	To exploit this vulnerability a malicious user must have access to at least Raspberry Pi.	Likelihood Classification	Medium
Risk control	Require use of secure credentials for access to Raspberry Pi; Restrict root access to the Raspberry Pi operating system; Enforce use of secure credentials for accessing the MQTT broker; Enable logs in Raspberry Pi;	Risk control classification	Acceptable
Residual risk	If access credentials are compromised the malicious user can get root access to Raspberry Pi.	Residual risk classification	Low

R.27	Increased attack surface due to unnecessary services running on the server	**Classification**	Low
Vulnerability	By default, the SO of the Raspberry Pi executes services that are not necessary for the operation of the prototype		
Impact evaluation	A malicious user can exploit this vulnerability to identify open communication ports and unnecessary services running on Raspberry Pi. Through communication ports or services, the malicious user can gain access to sensitive information or settings in the IoT solution.	Vulnerability	Medium
		Prior notice:	Yes
		Duration:	Short
		Impact (V+L+D)	Low
Likelihood	To exploit this vulnerability a malicious user needs access to the Wi-Fi network where Raspberry PI is installed and to know vulnerable ports or services.	Likelihood Classification	Medium

Risk control	Disable unnecessary services in the system; Implement security updates for the remaining services; Monitor the services running on the server;	Risk control classification	Efficient
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Residual risk classification	Low

R.28	*Physical damage due to weather conditions*	**Classification**	Medium
Vulnerabilit y	*Physical damage caused by temperature or humidity due to the installation location of the Raspberry Pi, sensor nodes and actuators*		
Impact evaluation	The occurrence of this vulnerability can affect the functioning of Raspberry Pi and, consequently, the IoT solution.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	Due to the characteristics of the environment where the IoT solution will operate, it is likely that the physical installation of Raspberry Pi will occur in locations subject to adverse weather conditions.	Likelihood Classification	Medium
Risk control	Install Raspberry Pi in a suitable location; Monitor server performance and temperature indicators;	Risk control classification	Acceptabl e
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Residual risk classification	Low

R.29	Database unavailable	**Classification**	Medium
Vulnerabilit y	The Firebase Realtime Database is hosted by a cloud provider and depends on the Internet to be accessed.		
Impact evaluation	In case of internet connectivity failure, the database will not be accessible.	Vulnerability	Medium
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium
Likelihood	Internet availability depends on the Internet provider and is subject to intermittent failure.	Likelihood Classification	Medium

Risk control	Implement backup for internet access; Implement mechanisms for off-line system operation;	Risk control classification	Acceptable
Residual risk	Redundant internet links raise the cost of operation. Off-line operation mechanisms can impact the ability to display messages in real time.	Residual risk classification	Medium

R.30	Exposure of the database on the Internet	**Classification**	Medium
Vulnerability	The Firebase Realtime Database is hosted by a cloud provider and may be accessible via Internet		
Impact evaluation	A malicious user can exploit this vulnerability to gain unauthorized access to the Firebase Realtime Database and access sensitive data stored in the database.	Vulnerability	High
		Prior notice:	No
		Duration:	Short
		Impact (V+L+D)	Medium
Likelihood	In order for this vulnerability to be exploited, access to one of the components of the IoT solution or through the Firebase Realtime Database administration console on the Internet is required.	Likelihood Classification	Medium
Risk control	Creation of a policy to access the database; Implement secure credentials to access the database;	Risk control classification	Acceptable
Residual risk	If credentials are violated the system is exposed; however, this risk can be mitigated by monitoring access to the database.	Residual risk classification	Low

R.31	Data may become corrupted	**Classification**	Low
Vulnerability	Data stored in the Firebase Realtime Database may become corrupted		
Impact evaluation	If this vulnerability occurs, the system data may become corrupted, impacting the operation of the solution.	Vulnerability	High
		Prior notice:	No
		Duration:	Medium
		Impact (V+L+D)	Medium

Likelihood	As the solution performs operations in real time, there is the possibility of data being corrupted. However, given the characteristics of the database this probability is low.	Likelihood Classification	Low
Risk control	Creation of a policy to access the database; Implement secure credentials to access the database; Monitor the database access;	Risk control classification	Efficient
Residual risk	The risks are reduced to a low level when implementing the proposed controls.	Residual risk classification	Low