A Fog Computing-based mechanism to provide reliability in IoT: A case study in a coffee smart farming environment

Master Degree work

Ana Isabel Montoya Muñoz

Advisor: PhD. Oscar Mauricio Caicedo Rendón
Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2021

A Fog Computing-based mechanism to provide reliability in IoT: A case study in a coffee smart farming environment

Ana Isabel Montoya Muñoz

Trabajo de grado presentado a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para obtener el título de:
Magister en Ingeniería Telemática

Advisor: PhD. Oscar Mauricio Caicedo Rendón

Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2021

Acknowledgements

First and foremost, I would like to thank God for keeping his promises and giving me the opportunities to grow up as a person and professional continually. I express my sincere gratitude to my mom, partner, family, and friends for their continuous support and love throughout the years. I want to dedicate this work to them. Furthermore, I would like to thank my outstanding academic advisor, Professor Oscar Mauricio Caicedo Rendón, for his support directing and guiding me for years with unconditional backing and helping solve all academic problems. Also, I would like to thank Jhonn, Carlos, David, and Rodrigo for all their support, encouragement, and accompaniment during these years.

Abstract

Reliability is essential in Smart Farming supported by the Internet of Things. A Fog Computing approach is pivotal for Smart Farming since it allows farmers to monitor and improve crop production by getting closer cloud capabilities at the edge of the network. The provisioning of reliability in farms is critical since the failure of a fog node can cause interruptions for farmers' decision-making services. Smart Farms' unprotection may cause significant economic losses and low yields of production. Moreover, making decisions based on inaccurate data can diminish the quality of crops and, consequently, lose money. This master dissertation addresses the Fog-based Smart Farms' unprotection from two approaches: system and data reliability. On the one hand, the dissertation introduces an optimization model for traditional protection schemes 1:1 and 1:N for meeting reliability in Smart Farms to minimize deployment cost to farmers giving heterogeneous fog nodes. On the other hand, we propose an IoT-Fog-Cloud architecture that incorporates a mechanism based on Machine Learning to detect outliers and another based on interpolation for inferring data intended to replace outliers. The proposed approaches were evaluated by conducting a case study in a network based on the proposed and deployed architecture at a Colombian Coffee Smart Farm. The results show the effectiveness of the proposed approaches regarding protection schemes in FN-based smart farms guaranteeing high reliability to improve the operation of farms; and high Accuracy, Precision, and Recall, as well as low False Alarm Rate and Root, Mean Squared Error when detecting and replacing outliers with inferred data.

Contents

List of Figures

List of Tables

List of Abbreviations

SF	Smart Farming
IoT	Internet of Things
FC	Fog Computing
FN	Fog Node
vSFF	virtual Smart Farming Function
SFFC	Smart Farming Function Chain
FDM	Failure Detection Mechanism
FRM	Failure Recovery Mechanism
ML	Machine Learning
FAR	False Alarm Rate
RMSE	Root Mean Squared Error
WSN	Wireless Sensor Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
IF	Isolation Forest
SVM	Support Vector Machine
eps	epsilon

Chapter 1

Introduction

The Smart Farming (SF) represents the application of Information and Communication Technologies (ICT) into the agriculture [1, 2]. SF aims to increase the quality of a product and the growth of agriculture yield with minimum human intervention. [3]. The particular environment of each crop affects directly the growth and productivity thereof. Therefore, it is necessary to monitor weather variables such as temperature, humidity, precipitation, and wind, to control quality, yield, traceability, and the spread of diseases [4, 5]. The manual collection of data and inspection of the crop status are error-prone, sporadic, unsupervised, and inaccurate, disturbing the farmer decisions [6].

The Internet of Things (IoT) has emerged as a suitable technology to collect and transmit several data from farms to users (*e.g.,* farmer, exporter, and consumer) because it is highly interoperable, scalable, open, and ubiquitous [7, 8]. IoT applications automatically offer farmers useful information to continually monitor the growth and weather variables of crops [9, 10]. The use of IoT platforms in SF (*i.e.,* IoT-based SF) will bring many benefits such as sensing systems, easier data exchange, device heterogeneity, data analytics, scalability, reasoning, and decision support [11]. Nevertheless, creating reliable wireless networking is still a significant challenge in IoT [12] since the computational power to process data is limited, the communication protocols consume too much energy, the Internet is not always available for IoT devices, and the topologies are dynamic which generate an error-prone

environment.

Reliability is the ability of a system to provide the correct service or the correct data over a period of time [13, 14]. In IoT-based SF, there is a lack of reliability due to failures in data collection and data transmission [15]. Some errors in data collection can be [16]: *i)* random errors by lack of sensor reading repeatedly, *ii)* spurious reading (*i.e.,* non-systematic reading errors) by a fake measure when some sporadic physical events happen (*e.g.,* if a camera flash triggers when measuring light intensity); and *iii)* systematic errors such as calibration, loading errors, and environment errors. Some data transmission failures can be: first, a sensor node failure causing a total or partial absence of data. Second, message omissions for sensor readings lost due to sensor failures and packet losses [17, 18]. Third, message delays when the timeliness of sensor data is a system requirement [19]. Fourth, a message corruption when the communication is disturbed by an unexpected fault such as natural disaster or if it has been intentionally attacked by a sensor manipulation [14].

The unreliability compromises the data and service quality that are necessary to avoid inefficiency in monitoring and control systems [20, 21] and to meet the farmer needs [22]. Thus, two of the main challenges in IoT-based SF are to assure the reliable data collection and transmission [23]. To detail the problem, let us consider a coffee farm as a particular SF scenario that involves IoT devices such as network-connected weather stations, low-cost computers, Wireless Sensor Networks (WSN), cameras, and smart-phones that collect a significant and heterogeneous amount of environmental and crop performance data [24]. These devices can measure: *i)* pH levels at the fermentation phase, *ii)* ambient temperature, air humidity, relative temperature and humidity of coffee beans at the dry phase, *iii)* temperature, time and color bean at the roasted phase; and *iv)* weather variables at any phase. These data are essential to guarantee the quality, taste, and aroma of coffee. Indeed, these data are the main input for applications such as data analysis to predict coffee production, traceability, and alarm systems [25, 26]. Therefore, the data collection and data transmission must be reliable to support the correct operation of the coffee farm and SF in general.

The concept of reliability has been studied in IoT at several domains. Some works

have proposed solutions to provide reliability in IoT by a self-control mechanism to the reliable measure of domestic conditions in-home monitoring systems [27], a time series-based prediction model to automate the reaction of a monitoring system [28], a distributed matching algorithm to manage tasks distribution between user nodes and Cloudlets [29], a smart IoT gateway to dynamic discovery and auto-registration of devices in-home-scale environments [30], and an aggregation scheme for Fog-based IoT to address the data privacy [31]. Nevertheless, none of the above works operates with a standardized concept of reliability. Then, there is not consistent way to provide reliability. Indeed, each work considers reliability from a different focus such as probabilistic, security, or self-management.

To sum up, to the best of our knowledge, the concept of reliability has not been studied deeply in SF. Thus, in the literature, a few works have considered reliable systems for SF. For instance, a self-control mechanism for environmental parameters monitoring and energy management [27]. An adaptive scheme for WSN nodes to increase the energy efficiency by undervolting [1] the specified minimum voltage levels of components on a potato field [32]. An application using Cloud Computing (CC) and Fog Computing (FC) to ensure farm animal welfare by managing production indicators [33]. It is noteworthy that these works present reliability results, but they do not introduce how such results were achieved. In fact, they do not propose a mechanism to provide reliability, nor they analyze the metrics that affect reliable systems (*e.g.,* energy consumption and packet losses).

Considering that IoT-based SF represents an appropriate solution in agriculture, this master thesis highlights that it is relevant the provisioning of reliability in data collection and transmission to afford services to the farmers that improve the quality of a product, the crops productivity, economic gains, and decision support. Provisioning of reliability includes the management of metrics such as devices density, link loss probability, and energy consumption. Therefore, this master thesis focuses on solving the following research question: **How to provide reliability in the collection and transmission of data in an IoT-based Smart Farming?**

In order to answer the raised research question, we present the following objectives.

[1] Operate in areas below the minimum specification of voltage.

General Objective

- Propose a mechanism for providing reliability in the collection and transmission of data in an IoT-based SF.

Specific Objectives

- Design a mechanism based on FC to manage reliability in the collection and transmission of data in an IoT-based Smart Farming.
- Implement a prototype of the proposed mechanism.
- Evaluate the reliability of the proposed mechanism considering network density, link losses, and energy consumption in a coffee Smart Farming environment.

1.1 Contributions and Scientific Production

We face the unreliability in IoT-based SF using FC. As farms are geographically far from cloud providers, FC offers the cloud capabilities needed by SF at the edge of the network, closer to the users or demand points. Moreover, FC is a suitable paradigm for constrained Internet connectivity scenarios such as farms, a common challenge in developing countries. It is essential to highlight that we address the provisioning of reliable smart farms by two approaches: system and data reliability. The first approach is oriented to find the optimal allocation (*i.e,.* placement of resources) of heterogeneous Fog Nodes (FNs) under reliability constraints. This model considers the protection of FNs by redundancy schemes. The second approach is directed to data cleaning mechanisms located at the Fog Tier. The Failure Detection and Data Failure Recovery mechanisms detect outliers and inferring data intended to replace them.

The investigation about providing reliability in IoT-based Smart Farms led to the following major contributions.

- An optimization model to minimize fog-based infrastructure deployment cost to farmers, considering heterogeneous nodes under reliability constraints.
- A comparison of redundancy protection schemes (*i.e.,* 1:1 and 1:N) for meeting reliability in SF.
- A Things-Fog-Cloud architecture that combines Machine Learning (ML) and Interpolation techniques to intelligently and automatically provide data reliability on SF applications.
- An ML-based mechanism for outlier detection in IoT-based SF data collection.
- An Interpolation-based mechanism for replacing the outliers detected with inferred data.
- A case study involving real data collected in a network based on a Things-Fog-Cloud and deployed in a Colombian Smart Coffee Farm.

The work presented in this monograph was reported to the scientific community by a paper published as a first author, another as a co-author, and one submitted to renowned journals (see Appendix B).

- **Ana Isabel Montoya Muñoz**, Oscar Mauricio Caicedo Rendón. **An approach based on Fog Computing for providing reliability in IoT Data Collection: A Case Study in a Colombian Coffee Smart Farm** Applied Sciences 2020.
  - Status: Published.
  - Special Issue: Computing and Artificial Intelligence.
  - Classification: A1 MinCiencias - Q2 (JCR).

- Jhonn Pablo Rodríguez, **Ana Isabel Montoya Muñoz**, Carlos Rodriguez Pabón, Javier Hoyos, and Juan Carlos Corrales. **IoT-Agro: A smart farming system to Colombian coffee farms** Computers and Electronics in Agriculture 2021.

  - Status: Published.
  - Classification: A1 MinCiencias - Q1 (JCR).

- **Ana Isabel Montoya Muñoz**, Rodrigo A. C. da Silva, Nelson L. S. da Fonseca, and Oscar Mauricio Caicedo Rendón. **Provisioning Protection to Smart Farming** Computers and Electronics in Agriculture 2021.
  - Status: Submitted.
  - Classification: A1 MinCiencias - Q1 (JCR).

Although not directly related to this master thesis, other peer-reviewed publication linked to the design of reliable network management solutions was published during the master.

- **Ana Isabel Montoya Muñoz**, Daniela Casas Velasco, Felipe Estrada Solano, Oscar Mauricio Caicedo Rendón, and Nelson L. Saldanha da Fonseca. **An approach based on Yet Another Next Generation for software-defined networking management** International Journal of Communication Systems 2021.
  - Status: Published.
  - Classification: A1 MinCiencias - Q3 (JCR).

1.2 Methodology and Organization

Figure 1.1 depicts the phases of the scientific research process that will guide the development of this master thesis: Problem Statement, Hypothesis Construction, Experimentation, Conclusion, and Publication.

- Problem Statement is to identify and establish the research question.

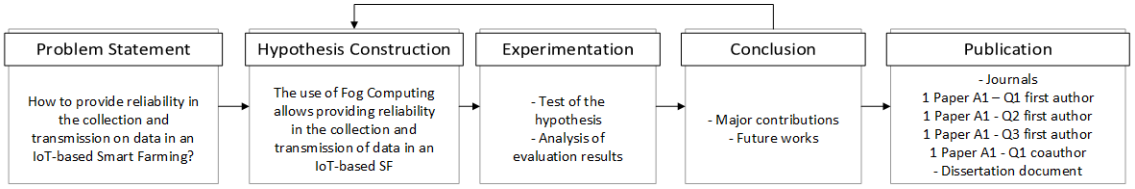


Figure 1.1: Master thesis phases

- Hypothesis Construction is to formulate the hypothesis and the associated fundamental questions. In addition, this phase aims to define and carry out the conceptual and technological approaches.
- Experimentation is to test the hypothesis and analyze the evaluation results.
- Conclusion is to outline conclusions and future works. Note that Hypotheses Construction has feedback from Experimentation and Conclusion.
- Publication is to submit and publish papers for renowned conferences and journals. The writing of the dissertation document also belongs to this last phase.

This master thesis document has been divided into chapters described below.

- Chapter 1 presents the **Introduction** that contains the Problem Statement, Objectives, Contributions and Scientific Production, and Methodology and Organization of this document.
- Chapter 2 presents the State-of-the-Art organized by the **Background** about the relevant topics concerning the performed research (including Smart Farming, reliability in IoT, and Fog Computing), and the **Related Work** that describes the researches works closer to the proposed approaches.
- Chapter 3 introduces the **Fog Computing-based system reliability approach in IoT** by the **Problem statement** and mathematical **Formulation** of the optimization model under reliability constraints, an **Evaluation** at several demand scenarios, and **Final remarks**.
- Chapter 4 introduces the **Fog Computing-based data reliability approach in IoT** based on **Reliable Fog Computing-based Architecture**, the **Data cleaning mechanisms to provide reliability**, and a **Case Study in a Colombian Coffee Smart Farm**.
- Chapter 5 presents **Conclusions** and **Future work**. The main conclusion of our work and outline future work are presented.

Chapter 2

State-of-the-Art

In this chapter, we present a background related to our approaches to provide reliability in SF. First, we describe SF, IoT, Reliability, and FC. Second, we review the related works according with the approaches of system reliability and data reliability.

2.1 Background

2.1.1 Smart Farming

The SF concept evolved from others paradigms such as Farm Management Information Systems (FMIS) [34] and Precision Agriculture (PA) [35]. FMIS studies planner systems for collecting, processing, providing and using all types of data necessary to support management operations of agricultural processes [36, 37]. PA covers the set of technologies that manage spatial and temporal variables of crop production [10] to contribute with long-term sustainability of production agriculture and improve the economics yields in an environmentally friendly way [38, 39]. Even if it is less used in the literature, another way to refer SF is Smart Agriculture (SA) [40]. SA is the integration of smartness into the agriculture using information technology that help farmers [41]. One of the main aims of making the agriculture smarter is the need to predict the demand for a crop so that the farmers are able to obtain fair

return of investment without wasting resources [42]. Nevertheless, SF still has some drawbacks such as the lack of high-quality information to meet the farm production requirements [43].

For this master proposal, the primary motive of SF is to provide a better solution for farmers to obtain a high yield, provide a high added value into the supply chain and improve food production processes by using ICT in agricultural production systems [1, 33]. SF involves the deployment of ICT into crops, equipment, machinery, and sensors; this generates a large volume of data and information that allows automating some processes such as monitoring and transportation [22]. SF relies on distributed systems around the farm that enable the combination and analysis of various farm data for decision making. Some authors suggest the integration of data mining [3, 44, 45] or artificial intelligence into farming to find correlations or patterns among the data collected (*e.g.,* weather sensors data) in order to generate decision support. For instance, an SF system alerts the farmers about critical weather conditions to take the best control strategies in the crops [25].

With the necessity to apply an integrated system that assures high productivity levels and monitoring systems at all stages of the agricultural value chain such as cultivation and harvesting [46], new technologies have emerged to increase the scope of SF, including IoT [22]. An IoT framework for SF offers various benefits such as deploying sensing systems, interoperability between different smart systems, easier data exchange from sensors, and increased automation by employing Internet standards [47].

2.1.2 Internet of Things

IoT is a novel paradigm that is rapidly growing in modern wireless telecommunications and monitoring scenarios [48] [49]. IoT has several definitions, for instance, it is defined as an infrastructure that will connect physical and virtual devices [49]. Also, IoT is outlined as the Internet that considers “things“ identified by unique addresses [50]. The network that connects devices having sensing capabilities is another IoT definition [51]. Furthermore, IoT has also been defined as the connection between

things that are dynamically configurable and communicates by interfaces-based on Internet protocols [52]. This master dissertation uses the IoT definition provided by [53], as a network that exchanges information to achieve intelligent identification, location tracking, monitoring, and management.

Traditionally, IoT widely uses the following technologies for the deployment of applications at different domains [54] [55]: *i)* Radio-Frequency Identification (RFID) that uses track tags attached to objects for identifying them automatically, *ii)* WSN that allows different network topologies and multihop communication between low-cost and low-power devices known as sensor nodes, *iii)* Middleware that provides services to make easier the communication for application developers; and *iv)* CC that provides hardware and software services from data centers over the Internet. WSN can be considered a specific implementation of IoT monitoring systems [56], particularly, using IoT devices as wireless sensors [57].

The IoT architecture for SF is divided into three layers from bottom-up [53, 58]: Sensing layer, Network layer, and Application layer. The Sensing layer includes IoT devices such as RFID tags, readers, cameras, sensors and so on, that are responsible for collecting data, recognizing objects, and perceiving the environment. The Network layer is in charge of all the data transmission, traffic management, routing technology, local processing, real-time monitoring and process control of the sensed data. The Application Layer process the data collection and transmission from the bottom layers to deploy services to end-users through communication interfaces. An example of their services is the prediction of coffee production based on weather parameters.

2.1.3 Reliability in IoT

Since there is not a standardized definition of reliability in IoT [59], this section presents the diverse definitions of reliability found in the IoT literature. Furthermore, this section exposes a more specific definition of reliability in WSN technology which is an essential component of IoT in SF.

In IoT, the reliability has several definitions. For instance, the ability of a system or

component to perform its required functions under stated conditions for a specified period of time [60]. The probability that a program will not cause the failure of a system for a specified time [61]. The ability of a system to prevent itself from failing [28]. The ratio of calculated value with the information received correctly [62]. The performance of components such as radio transceiver and communications micro-controller [12]; and in terms of availability [30, 32] or security [31].

In WSN, [21] introduces a reference model to classify reliability. This model establishes a common taxonomy to evaluate the research in WSN reliability involving different combinations of retransmission or redundancy techniques that aim to ensure event or packet level reliability by using either end-to-end or hop-by-hop methods [23]. The following WSN reliability techniques and methods consider both data collection and data transmission.

- *Retransmission-based reliability* is the most commonly used technique to achieve data transmission reliability. With this technique, the sender node waits for the acknowledgment of the next hop node on the path after transmitting its packet.
- *Redundancy-based reliability* aims to correct only the lost or corrupted bits within the packet. The sender adds some extra information to the packet that the receiver can use to reconstruct the packet if some bits are lost or corrupted.
- *Event level reliability* aims to ensure that the sink [1] only gets enough information about a certain event happening in the network instead of all the sensed packets.
- *Packet level reliability* aims to ensure that all the packets carrying sensed data from all the related sensor nodes are reliably transported to the sink.
- *End-to-end* is a connection-oriented mechanism, where only the two end points (*i.e.,* only the source and destination nodes) are responsible for ensuring reliability.

[1] Base station used to collect and process data in centralized mode.

- *Hop-by-hop* is a link-oriented mechanism, where every single hop from source to destination is responsible for ensuring the reliable transmission of the sensed data.

Related to the reliability in WSN, there is a problem of data quality. In [14], the authors provide a comprehensive overview of the solutions to achieve improved sensor data quality and reliable operation of WSN-based monitoring applications. They highlight the following different fault detection methods to provide reliability based on data quality:

- Rule-based methods that use expert knowledge about the variables that sensors measure to determine operation thresholds of sensors.
- Estimation-based methods that define a "normal" behavior by considering spatial and temporal correlations from sensor data.
- Learning-based methods that define models for correct sensor measurements, using collected data for building the models.

Redundancy techniques have also been extensively utilized to enhance the reliability of engineering systems [63]. A redundant system is a system that remains in stand-bay a set of nodes to guarantee its normal operation. The redundant nodes are configured in parallel as backups; then as long as not all of the system components fail, the entire system works. In other words, the total system reliability is higher than the reliability of any single system component. Figure 2.1 shows the redundant system reliability computation of "n" nodes where Rs is the total reliability and Ri is the reliability of each component.

Another standard method to compute reliability is by the Mean Time Between Failure (MTBF) metric. Even though MTBF and reliability are different, MTBF can be converted to reliability by using the following equation for exponential distributions where e is the epsilon mathematical constant (*i.e.,* 2.71828), t is the time in hours of operation, and the MTBF is also expressed in hours. For instance, to calculate the reliability at 8760 operation hours (*i.e.,* one year), Equation 2.1 indicates that

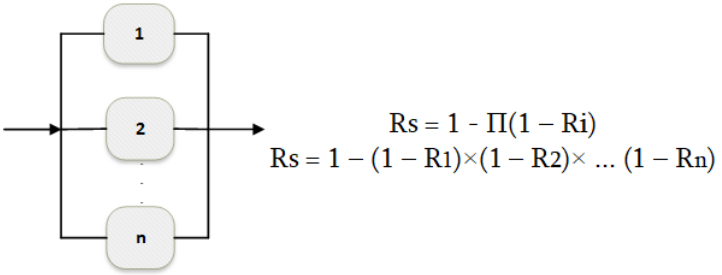


Figure 2.1: Redundant system

the probability that a system with an MTBF of 26280 hours (*i.e.,* three years) will still be functioning after the operation time, is the 71.6%.

$$R(t) = e^{-\frac{t}{MTBF}} \tag{2.1}$$

SF is an error-prone environment needing system reliability since the IoT infrastructure deployment is outdoor, implying possible outages for lack of coverage, crashes, or loss for harsh environmental conditions [64]. Moreover, making decisions based on inaccurate data can diminish the quality of crops and, consequently, lose money. We address these dependable challenges by two approaches: system and data reliability. For this master thesis, system reliability is the ability of a system to provide continuity of correct service over an operation time, and data reliability is the ability of a system to give accurate data over a period of time [13, 14].

2.1.4 Fog Computing

In the literature, there are diverse concepts to refer to the intermediate layer between Things and Cloud. Originally, this concept was known as Edge Computing (EC) [65] or Edge Computing Technologies (ECT) [66]. EC offers real-time response by bringing networking and computational resources on edge devices (*e.g.,* routers, gateways, switches, and base stations) near to the end user. EC include emerging technologies such as FC, Mobile Edge Computing (MEC), Micro Data Centers (MDCs), and Cloudlet. The primary aim of FC is to incorporate virtualized services

into the network edge and network devices, in terms of processing, storage, and networking services [67, 68, 69]. The major purpose of MEC is Radio Access Networks (RANs) in 4G and 5G cellular networks. MEC offers EC by proposing a placement of computation and processing resources at base stations [70, 71]. However, its scope has been expanded in the last year to encompass non-mobile network requirements, replacing the "Mobile" by "Multi-Access" [72]. MDCs are the reduced scaled version of data centers, which extend the offered services of the Cloud near to the end users [73, 74, 75]. Cloudlet is a small virtualized "data center in a box" close to the mobile devices to serve edge users in a distributed way [76, 77].

In this master thesis, we are going to study FC [78] because Fog Nodes (FNs) [79] define devices with less storage resources and more energy limitations than Cloudlets and MDC, which meets the resource constraints of SF. FC provides services for computing, storage, and networking between end-devices and traditional Data Centers [80]. FC addresses the limitations of CC when the number of connected devices increases and compromises the Data and Service Quality, especially for delay-sensitive applications [81]. The term FC was introduced by Cisco as a highly virtualized platform suitable for a number of critical IoT services and applications [82]. Other authors define FC as a distributed computing paradigm that fundamentally extends the CC paradigm from the core to the edge of the network [72, 83, 84], and as a "cloud closer to ground" [85].

FC has the following main characteristics [69] [84] [86]: *i)* location awareness of distributed FNs, *ii)* lower latency of end-devices data processing, *iii)* geographical distribution of services and applications provided by the Fogs deployed anywhere, *iv)* scalability by providing distributed computing and storage resources that work with large-scale sensor networks, *v)* support for mobility by the ability to connect directly to mobile devices and to enable mobility methods, *vi)* real-time processing between FNs, *vii)* heterogeneity of FNs or end-devices designed by different manufacturers that need to be deployed according to their different platforms, *viii)* interoperability by working with different domains and across different service providers, *ix)* support for online analytic and interplay with the Cloud by processing data close to end-devices; and *x)* reduction of operating expenses by processing selected data locally instead of sending them to the Cloud for analysis.

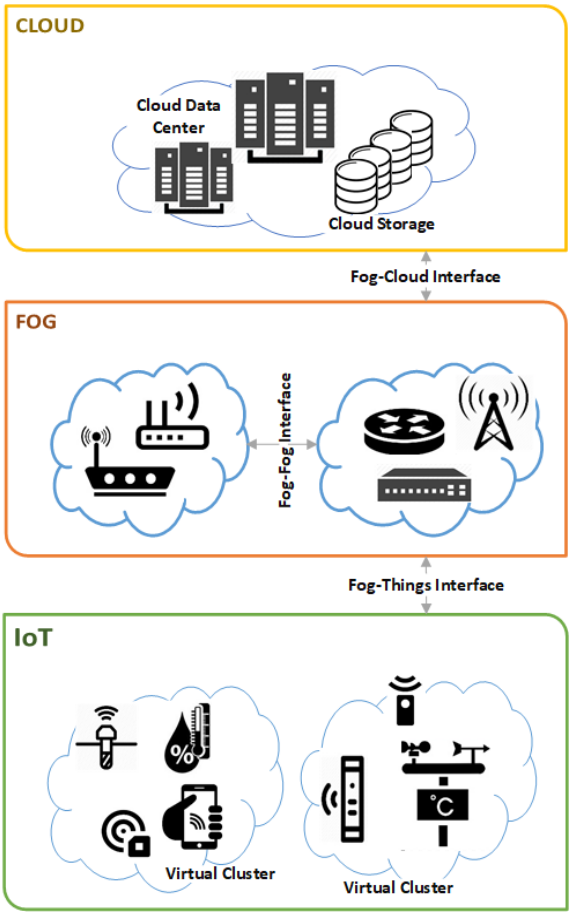


Figure 2.2: Fog Computing Hierarchical Architecture

Figure 2.2 depicts the Fog hierarchical architecture that is formed by a Cloud Tier at the top, a Fog Tier at the middle and an IoT Tier at the bottom [87, 88, 89]. The Cloud Tier consists of one or more traditional DCs that provide Infrastructure as a Service (IaaS) [90], Platform as a Service (PaaS) [91], and Software as a Service (SaaS) [92, 93]. The Fog Tier is composed of FNs. FNs are physical or virtual entities (*e.g.,* routers, switches, low-cost computers, wireless access points, video surveillance cameras, controllers and servers) that communicates end-users and DCs, aimed to ease the execution of IoT applications [94]. This tier can be divided into multiple ones according to the requirements of applications as a Multi-Service Edge Tier. It is important to highlight that FNs can collaboratively share storage and computing installations, and can co-exist with network elements reusing the wireless interface. The IoT Tier or Things Tier is responsible for receiving/sending data, performing

a first stage analysis, and passing its results to the intermediate Fog Tier for more detailed analysis [95]. The IoT Tier is formed by IoT devices such as sensors, wearables, embedded systems, and so on. These devices can interact with the upper layers and with each other too.

The fog network can be empowered by Software Defined Networking (SDN) and Network Functions Virtualization (NFV) [96] to assure the dependability of IoT integrated system, meet the QoS requirements of the fifth-generation (5G), managing and controlling smart devices. The SDN paradigm decouples the control plane from the data plane by offering a centralized logic, a programmatic configuration, and open standards-based protocols [97]. [98] states that "some gateways can be deployed as virtual machines and their traffic can be tightly controlled thanks to SDN capabilities in a local edge cloud". Thus, SDN and NFV make possible the placement of controller on the end nodes of fog network, improving the services by the virtualization of gateways, switches, load balancers, firewalls, and other functions on those FNs [99].

Since the benefits of SDN and NFV were considered as a possible component of a solution in FC-based IoT for providing SF reliability, the research carried out in the master included SDN-integrated management in a technology-agnostic and heterogeneous environment. The results about SDN-integrated management were published in a paper contributing to this master thesis (Section 1.1). The findings of that paper were not included in the core of this thesis because the (actual) farm deployment used in the case study is not SDN-based. Nonetheless, SDN and virtualization concepts were incorporated in the proposed reliability approaches, such as functions virtualization, resources allocation, and hierarchical controller-based architecture.

2.1.5 Fog Computing for Smart Farming

Fog can control and manage IoT-based SF environments aiming at meeting the requirements related to the limited capacity for storage, energy constraints and computation of sensors [8, 100]. Some works have used FC to save bandwidth and locally process data such as bonsai moisture measurements [101] and to ensure farm

animal welfare (cows and pigs) by managing temperature, GPS, cameras, humidity, and accelerometer [33]. Furthermore, FC can improve other SF applications such as coffee production, traceability, farm data security, and farm alarm system.

Thanks to its advantageous features, Fog can provide reliability in IoT-based SF by managing constraints of latency, network bandwidth, device resource and energy, services with intermittent connectivity to the Cloud, and IoT security [102]. Moreover, reliability is one essential research area in FC [103]. Fog can provide reliability into the interoperability issue [104] caused by the variable traffic generated by IoT devices. FNs can handle various types of data collected, providing an environment where heterogeneous devices can operate coherently.

2.2 Related Work

2.2.1 System Reliability in IoT-based Smart Farming

Table 2.1 summarizes research works on system reliability, which considers the involvement of SF, IoT, Edge or FC and the objective function if the work uses optimization. Recall that reliability is a standardized measure in software engineering disciplines, but there is no agreed definition for IoT [105], much less for SF. For this reason, we review principles of reliability in other domains.

Regarding SF domain, [27, 32, 106, 107] highlighted the relevance of reliability in SF scenarios in terms of receiving sensor data correctly, saving energy by operating sensor below the minimum specification of the voltage, comparing measurements between generic and specialized sensors, and as a coefficient regarding rainwater harvesting system. Most of the works aforecited handles reliability as a system qualitative feature and not as a quantitative evaluation parameter. Moreover, none work used traditional techniques, such as redundancy, nor implemented an FC-based approach to provide reliability.

Several works have addressed the reliability in an edge computing domain. The work [33] considered an SF scenario using FC for managing an animal welfare ap-

plication locally. But this work proposed only a local database in case of failures to handle data reliability. [108] proposed a fault-tolerant mechanism by combining Directed Diffusion and Limited Flooding to enhance the data transmission reliability. This mechanism was designed to evaluate the health status of older adults by using a Reduced Variable Neighborhood Search based on the sensor Data Processing Framework. [31] proposes a reliable and privacy-preserving selective data aggregation scheme for the fog-based IoT. This paper defines the non-collusive and collusive attacks for the particular data aggregation service. Thus, the authors handle reliability, stating that the data should not be forged/modified and intact.

For the SDN-based FC (SDN-F) domain, [96] predicts the current reliability of legacy links based on their previous reliability values at the SDN controller using an improved K-NNR algorithm. [109] estimates the link reliability level using an ML algorithm. These works model path reliability in objective functions for SDN-enabled IoT-Fog to improve network management. However, none of these optimization models are related or applicable to an agriculture scenario which is the domain addressed in this master thesis.

[110] and [111] present optimization solutions for fog-based location-allocation problem. The model in [110] evaluates two types of demands: strict (which can only be processed in a FN) and flexible (which can be hosted either in the fog or in the cloud). The goal is to process most of the strict workload in the FNs using the minimum number of servers possible to reduce the overall cost. The authors do not include reliability constraints in their model. [111] introduces the PACK algorithm that optimizes the placement of a fixed number of edge servers, minimizing the distances between servers and Access Points (APs) while balancing system workload and satisfying server capacity constraints. The PACK algorithm only considers reliability by providing reserve capacity for APs that are a critical part of the network infrastructure.

In particular, for Edge Computing-based 5G domain, [29] proposed a proactive edge caching of Ultra-Reliable and Low-Latency Communication (URLLC) services in Fog networks. The reliability consists of sending the same requests to various servers, and using whatever response comes back first. This approach minimized the total

Table 2.1: Related work - System reliability

Ref	Domain	Reliability Definition	SF	IoT	Edge/Fog	Optimization Objective
[32]	Agriculture	Availability over operating areas below the minimum specification of voltage	✓			
[27]	WSN Homes	The high ratio of theoretical value with the sensor information received correctly	✓	✓		
[106]	Agriculture	Low reading difference between generic meter and specialized sensors node	✓	✓		
[107]	Greenhouse	The ability of the rainwater tank to satisfy the water needs of tomato and begonia crop	✓	✓		
[33]	Animal welfare	Replicas of the database in case of failure	✓	✓	✓	
[108]	Healthcare	Fault-tolerant data transmission		✓	✓	
[31]	Fog-based IoT	Integrity of data		✓	✓	
[109]	SDN	Link reliability as the average failure/-downtime or repair time in specific time duration		✓	✓	Maximize the path reliability and minimize the path delay
[96]	SDN-F	Edge's failure and downtime probability		✓	✓	Maximize path reliability,and minimize delay and bandwidth utilization
[110]	5G	No formal definition		✓	✓	Maximize the processing of strict workload in the FNs using the minimum number of servers possible to reduce the overall cost
[111]	City-wide public Wi-Fi network	Replication of a few critical Wi-Fi Access Points		✓	✓	Minimize the sum of weighted distances between the edge servers and the APs taking into consideration the workload of each AP and the capacity constraints of each server
[29]	5G	A probabilistic constraint on the maximum offloaded computing delay		✓	✓	Minimize the total task computing latency under reliability constraints
[112]	5G	The ability of a system to provide service continuously over a defined period of time			✓	Minimize the number of MEC deployed, the number of hosted slices, and the total service response time
[113]	5G	The ability of a system to provide service continuously over a defined period of time			✓	Minimize overall service provisioning cost, service response time and service loss probability
Chapter 3	Agriculture	The ability of a system to provide continuity of correct service over an operation time	✓	✓	✓	Minimize fog-infrastructure cost under reliability constraint

task computing latency under reliability constraints, by efficiently distributing and proactively caching the results of the computing tasks.

[112] highlighted that reliability can be assured by dynamic resource provisioning at the edge nodes to cope with failures. This work introduced a Multi-access Edge computing location problem, which aims at selecting locations to place edge nodes hosting protected slices to meet the URLLC requirements. The authors evaluated traditional protection schemes to provide reliability for latency-stringent services. [113] introduced a capacitated reliable facility location problem with failure probability for edge device placement and reliable Broadcasting in 5G NFV-based small cell networks. In the problem formulation, a standby backup facility will serve the demands hosted by another facility if the primary facility fails or if the facility's load reaches its capacity. The authors employed an extended genetic NSGA-II algorithm to locate edge devices to provide reliable broadcast services to minimize the service response time, service loss probability, and service provisioning cost. Although [29, 112, 113] provides and evaluates reliability at the edge of the network, do not proposes a protected FNs schemes involving SF scenarios to minimize cost.

As far as we know, no work proposes protected SFs considering Fog layer deployment's reliability and cost for farmers' decision-making applications to perform dependably. The system reliability approach presented in Chapter 3 introduces, first, a model for optimizing the resource allocation of FNs serving farm's demand points in terms of processing and memory that seeks to minimize implementation costs under reliability constraints. Second, two schemes to provide fog layer protection by redundancy: one o more backups for each primary FN (1:1 and 1:N). Third, new concepts to model common farming functions, consider heterogeneous FNs and different types of demands.

2.2.2 Data Reliability in IoT Collection

Table 2.2 summarizes research works on data reliability, which considers the involvement of IoT, FC, SF, mechanisms for outliers detection, and mechanisms for data recovery when outliers happen.

The works in [27, 28, 30, 32, 114, 115, 116, 117] introduced IoT applications, mostly in SF domains. These works focus on deploying specific applications such as monitoring for greenhouses, a control system for watering crops, and a system to minimize the use of fertilizer and pesticides. Although they highlight reliability as an essential feature for IoT-based SF, they did not design their solutions from a reliability perspective. [29, 31, 108] managed reliability in FC-based IoT by using optimization, self-adaptability, and re-transmission techniques. Nevertheless, none of these works included failure data detection and data recovery mechanisms to provide reliability in SF scenarios. [33, 7, 118, 119, 120, 121] introduced novel IoT-based SF applications using FC features, such as local processing, deploying control modules near the access network, minimizing the amount of data transferred to the cloud, and reducing delay. These works did not exploit the FC capabilities for data reliability improvement. [122, 123, 124] introduced mechanisms to detect anomalies in datasets. However, none of these works compared two or more methods for anomaly detection nor considered an SF environment based on FC and IoT. [125] and [126] proposed mechanisms for detecting and handling outliers by interpolation, and an algorithm to remove spatial outliers in high-density data sets but this work did not consider FC-and-IoT based-SF.

There is no approach based on ML and interpolation techniques, to the best of our knowledge, aimed to provide data reliability to IoT-Fog-Cloud-based SF applications that support decision-making to farming stakeholders. Our data reliability approach presented in the Chapter 4 seeks to address the shortcomings mentioned above by introducing a Fog-based and reliability-oriented architecture that incorporates a mechanism for detecting outliers in data and another mechanism for handling them to offer reliable data. Also, it is to highlight that our approach operated with data collected in a real SF scenario.

Table 2.2: Related work - Data Reliability

Ref	Description	Reliability Definition	IoT	FC	SF	Outlier Detection	Data Recovery
[28]	A system architecture for monitoring the reliability of IoT	The ability of the system to prevent itself from failing by continuously introspecting its state and take decisions without human intervention	✓				
[30]	A self-configurable IoT gateway for interconnecting non-IP devices in small-scale IoT environments	Availability and self-configuring	✓				
[114]	An optimal system watering agricultural crops based on a Wireless Sensor Network (WSN)	No formal definition	✓		✓		
[115]	A micro-climate monitoring and control system for greenhouses	Mean-battery life and network mean packet reliability	✓		✓		
[116]	A methodology for data cleaning to eliminate yield data errors in Winter cereals yield tracking system	The correct data collection in yield maps	✓		✓	✓	
[27]	A new framework for monitoring IoT systems intelligently	The high ratio of theoretical value with the sensor information received correctly	✓		✓		
[32]	An adaptive and reliable undervolting scheme for WSN nodes	Availability over operating areas below the minimum specification of voltage	✓		✓		
[117]	A detailed framework to cater full fledged agricultural-solutions using IoT	No formal definition	✓		✓		
[29]	A proactive computing and task distribution scheme for ultra-reliable and low-latency fog computing networks	A probabilistic constraint on the maximum offloaded computing delay	✓	✓			
[31]	A reliable and privacy-preserving selective data aggregation scheme for Fog-based IoT	Integrity of data	✓	✓			
[108]	A framework for reliable data transmission in a healthcare IoT system supported in Fog computing	Fault-tolerant data transmission	✓	✓			
[33]	A framework for developing and deploying on animal welfare applications	Replicas of the database in the Cloud, in case of failures	✓	✓	✓		
[7]	An IoT platform to face soilless culture needs in full recirculation greenhouses using moderately saline water	The ability to avoid network access failures	✓	✓	✓		
[121]	An IoT system with fog assistance and cloud support that analyzes data generated from wearables on cows	Availability	✓	✓	✓	✓	
[118]	SmartHerd, an IoT platform solution that addresses the connectivity and animal welfare in a smart dairy farming scenario	Availability	✓	✓	✓		
[119]	An IoT application that monitors the cattle in real-time and identifies lame cattle at an early stage	Accuracy	✓	✓	✓		
[120]	An IoT platform for agricultural monitoring automation, and pest management image analysis	No formal definition	✓	✓	✓		
[122]	A mechanism for discovering anomalies in a temperature dataset using the DBSCAN algorithm	The ability to detect outliers				✓	
[123]	An anomaly detection framework for streaming data	The ability to detect outliers				✓	
[124]	An in-network knowledge discovery approach for detecting outliers in sensor networks	Degree of truthfulness of the readings obtained from each sensor				✓	
[125]	A mechanism that handles outliers via an interpolation method based on neural networks	Data Quality				✓	✓
[126]	An algorithm to identify and exclude spatial outliers in the high-density spatial data set.	Data Quality			✓	✓	✓
Chapter 4	An approach for providing reliability in Fog-based IoT	The ability of a system to provide the correct data collection in Smart Farming data over a period of time	✓	✓	✓	✓	✓

Chapter 3

A Fog Computing-based System Reliability Approach in IoT

3.1 Introduction

SF represents the application of ICT into agriculture [2]. SF aims at increasing the quality of a product, and the growth of agriculture yield with minimum human intervention [3]. An FC approach involving an IoT-Fog-Cloud continuum is pivotal for SF since it allows farmers to monitor and improve crop production by data analytics [10, 47, 117]. As farms are geographically far from cloud providers, FC can be used to offer at the edge of the network the cloud capabilities needed by SF [27, 33, 127, 128]. FC aids to address some SF challenges, such as connectivity and availability constraints and limited capacity for storage and processing of the sensors [8].

Despite the FC advantages, the provisioning of reliability in SF is critical since the failure of an FN can cause interruptions for farmers' decision-making services. Remarkably, Smart Farms' unprotection may cause significant economic losses and low yields of production [11, 12, 129]. For example, if the server that transfers essential data to guarantee taste, value, and aroma of coffee [24] goes down, the coffee production will not meet the quality standards; hence it will not be acceptable to

exportation. Moreover, SF is an error-prone environment since the device's deployment is outdoor, leading to possible outages for lack of coverage, crashes, or loss for harsh environmental conditions. Ensuring reliability in resource-constrained IoT networks is one of the primary concerns to achieve a high degree of efficiency in monitoring and control systems [21].

For provisioning protection to SF is necessary to achieve devices continuum operation without fault over a given time. This chapter introduces a novel model for protecting Smart Farms by the Fog Layer infrastructure reliable deployment (protected FNs by redundancy) for providing services to farmers minimizing the implementation costs. We introduce the following new concepts to model the proper resource allocation in protected and heterogeneous FNs regarding processing and memory: virtual Smart Farming Functions (vSFFs) and Smart Farming Function Chains (SFFCs), referring to an algorithm that performs a function related to a farm service, and the chaining of these functions, respectively. Furthermore, we evaluate the following schemes to provide reliability to SF services:

- $1:1$, a dedicated backup node is reserved for each service demand.
- $1:N$, a backup node is shared among N potential primary nodes.

3.2 Problem Statement

The approach introduced in this chapter considers Smart Farms based on a traditional FC architecture formed by Cloud, Fog, and Things layers (Figure 2.2). The Cloud layer includes one or more DC with high processing, memory, and bandwidth resources. The Fog layer is composed of FNs. FNs are physical or virtual entities (*e.g.,* wireless access points, raspberry pi, controllers, and servers) that facilitate the communication between end-users and DCs, these entities are intended to ease the execution of IoT applications [94]. The Things layer comprises devices that sense and transmit data to the FNs for processing [95].

The resources of FNs can be shared to meet demands points from multiple farm services. A demand point represent a service request from farms stakeholders. A farm

service (*e.g.,* crop monitoring and automatic irrigation) corresponds to an SFFC that has different processing, memory, and latency requirements. FNs can process several SFFCs on-demand according to the available capacities for supporting the requirements mentioned above. A set of vSFFs forms an SFFC. A vSFF is a primary function needed to perform farm applications.

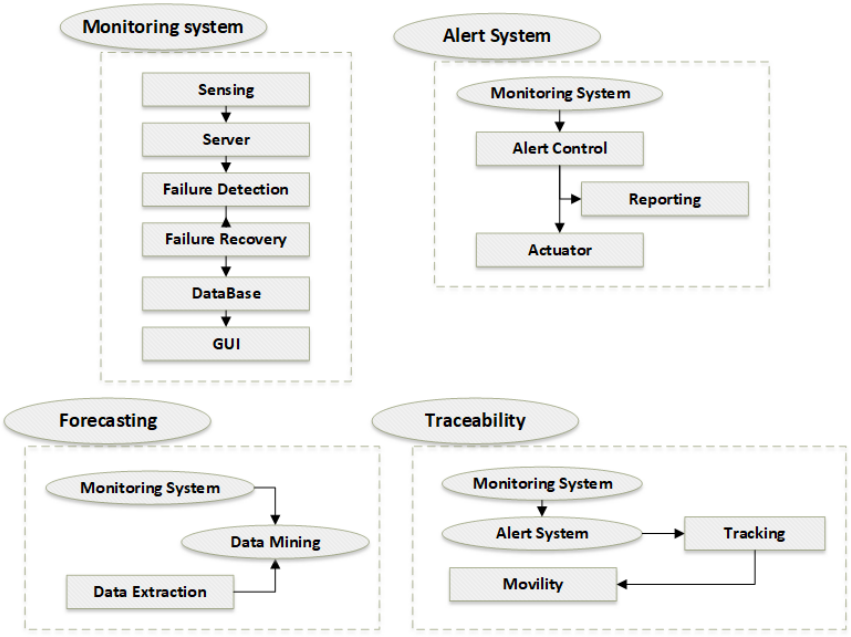


Figure 3.1: virtual Smart Farming Functions

Figure 3.1 depicts traditional vSFFs, such as collecting temperature values from a sensor or saving data in a database. Note that a farm service can be composed of another farm service. For instance, the alarm system contains all the vSFFs from monitoring plus the vSFFs necessary to control, report, and act. Let us consider that an $SFFC_1$ is the corresponding chain from the service of forecasting and $SFFC_2$ for traceability. Each service demands particular requirements regarding processing and memory. All vSFFs of a chain are hosted by the same primary FN. In case of failure, the whole chain is transferred to a backup device.

The goal is to design a model able to find the optimal allocation of SFFCs in terms of CPU and RAM in FNs to meet the users' demand points minimizing deployment costs. By allocation, we mean the assignment of SFFCs demanded by the users, which defines the FNs activation. Allocation aims to locate the resources in a way

that supplies the demand points most efficiently. In addition, the model envisions the redundancy of FNs by finding optimal assignation of backups under reliability constraints and comparing two protection schemes: 1:1 and 1:N. The model also differentiates two levels of protection assignment (*i.e,*. 0 - primary node, and 1 - secondary or backup node). Moreover, it distinguishes three types of FNs, from 0 - lowest resources to 2 - highest resources.

Figure 3.2 illustrates the $1:1$ protection scheme, in which the primary FN contains SFFCs for specific farm service, and a dedicated backup is assigned in a different FN of the same type when the primary one fails. For instance, the primary FN_1 of type 0 process the SFFC composes of $vSFF_1$, $vSFF_2$, and $vSFF_3$; in case of failure, the dedicated backup FN_2 of type 0 will be in charge of the same SFFC.

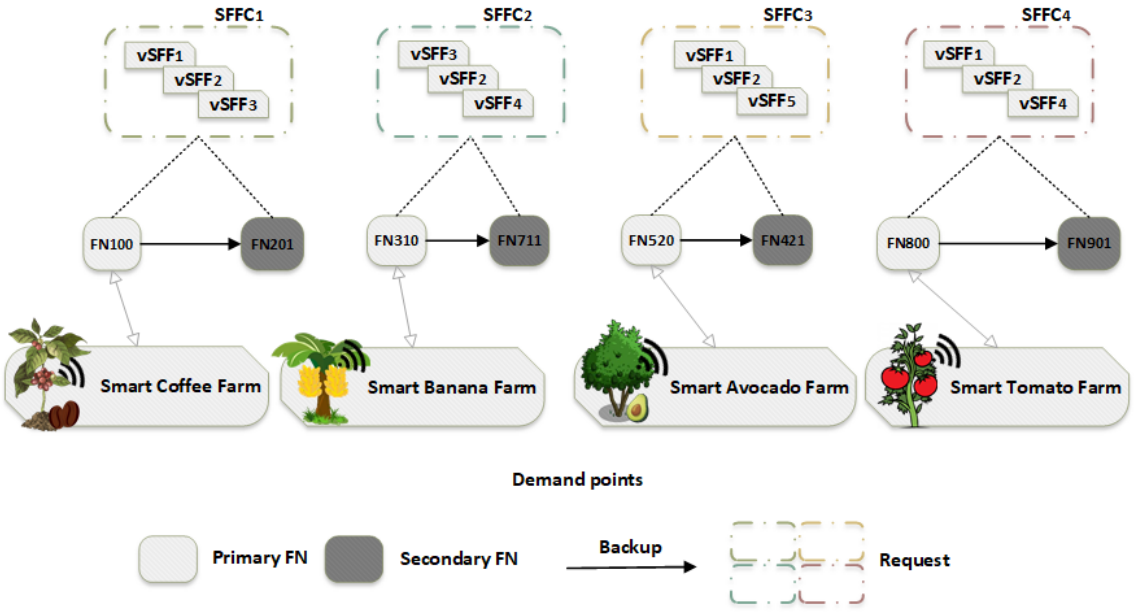


Figure 3.2: Dedicated Scheme $(1:1)$

Figure 3.3 presents the $1:N$ protection scheme, in which the primaries FNs contain SFFCs for a specific farm service, and there is a potential backup FN (of the same or higher type) for the N assigned primary nodes in case one fails. The secondary FN can replace N potential primary nodes depending on the capabilities of memory and processing. For example, the primary FN_1 of type 0 process a SFFC composes of $vSFF_1$, $vSFF_2$, and $vSFF_3$, and the primary FN_3 of type 1 process a SFFC composes

of $vSFF_1$, $vSFF_2$, and $vSFF_5$. For a $1:2$ scenario, the secondary FN_5 of type 2 is the potential backup of the SFFCs of FN_1 and FN_3 in case of one fails.

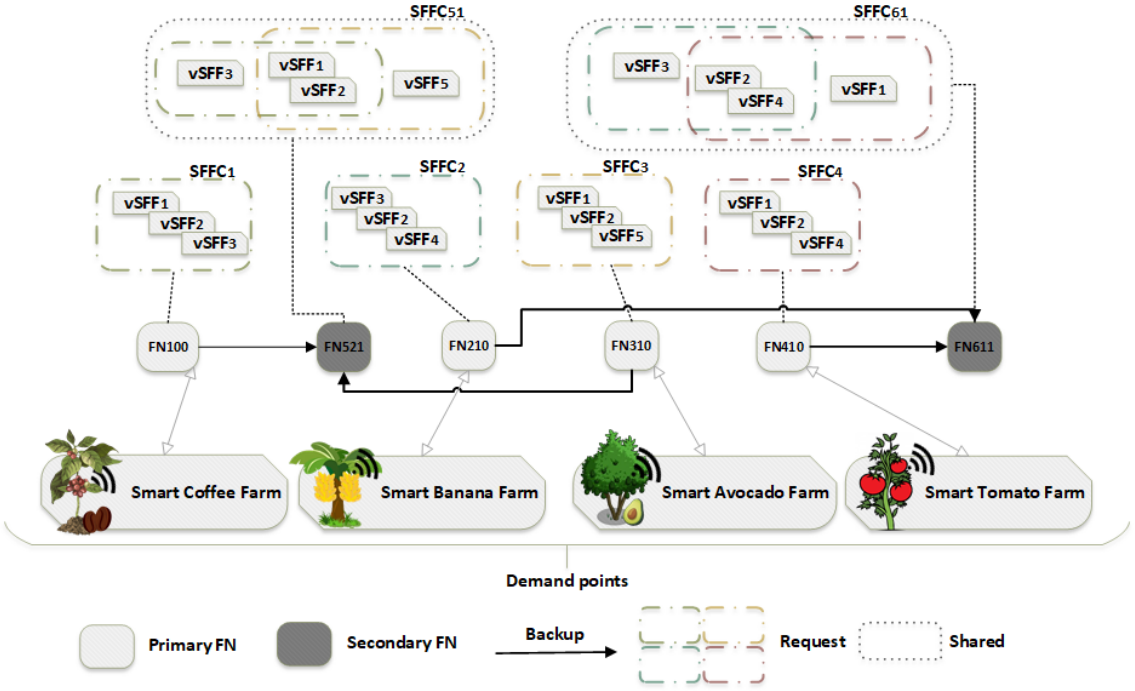


Figure 3.3: Shared Scheme $(1:N)$

3.3 Problem Formulation

This section introduces the formulation of heterogeneous FN allocation problem with two protection schemes. The goal is to devise the optimal number of FNs (hosting SFFCs or vSFFs) based on available resources (*i.e.*, processing and memory) and reliability requirement to provide seamless connectivity for farms services and minimizing deployment costs. The notation used in the formulation is summarized in Table 3.1.

Let $G = (U \cup V, E)$ be a bipartite graph in which U denotes the set of potential FNs where the vSFFs of farming services can be activated to serve the requests from the demand points (users from several farms) $v \in V$. The network link between the FNs and demand points is defined by $E \subseteq U \times V$. FNs are heterogeneous. We

Table 3.1: Notation used in the FN allocation problem formulation

Symbol	Description
U	Set of FNs
V	Set of demand points
I	Index of FN type (i=0 - lowest resources, i=2 - highest resources)
f_{ui}	Fixed cost to activate a FN $u \in U$ of type i in USD
K	Level of protection assignment of a FN (k=0 - primary, k=1 - secondary)
Ψ_{ui}	CPU capacity of FN $u \in U$ of type i in MIPS
Φ_{ui}	RAM capacity of FN $u \in U$ of type i in GB
τ_v	Processing (CPU) requirement of a demand point v assigned to a FN $u \in U$ in MIPS
σ_v	Memory (RAM) requirement of a demand point v assigned to a FN $u \in U$ in GB
r_u	Reliability of a FN at the location $u \in U$
R_v	Required reliability level of a demand point $v \in V$

differentiate three types of FNs according to CPU (Ψ) and RAM (Φ) capacities, reliability rates, and cost.

FNs can fail for several reasons, such as unexpected restart and shutdown, especially for power outages often in developing countries' farms. Each FN $u \in U$ is associated with a reliability value, r_u, which defines its probability of no-failure operation for a given operation time. The FNs hosting a demand point are categorized either as primary or as secondary (*i.e.,* $k = 0$ primary node, $k = 1$ secondary node), depending on its role in protecting. If a primary FN fails, a node assigned as a secondary (backup) hosts the demands of the failed node.

The objective is to find optimal allocation of FNs hosting vSFFs that compose a farm service requested meeting reliability while minimizing deployment cost. We address this objective by formulating an FN location problem extended with protection schemes and cost minimization, which finds: the optimal number of FNs to minimize monetary cost given heterogeneous FNs under two protection schemes ($1 : 1$ and $1 : N$).

The solution for the FN location problem is given by a objective criteria formulation which employs the binary decision variables in Table 3.2.

Table 3.2: Decision variables

Notation	Description
$y_{uik} \in \{0,1\}$	Value 1 indicates that the FN $u \in U$ of type i as k assignment ($k = 0$ primary node, $k = 1$ secondary node) is active
$x_{uv} \in \{0,1\}$	Value 1 indicates that the FN $u \in U$ serves a demand point $v \in V$
$z_{uw} \in \{0,1\}$	Value 1 indicates that the secondary node $w \in U$ backing up the primary node $u \in U$ is active

The objective formulation has the following objective function:

$$Min \sum_{k \in K} \sum_{u \in U} \sum_{i \in I} f_{ui} y_{uik} \tag{3.1}$$

The objective function defined by Equation 3.1 aims at minimizing the cost to active heterogeneous FNs. The above can be interpreted as: the higher the number of activated FNs (primary and backups), the greater is the cost incurred.

The constraints of the problem are the following:

$$\sum_{u \in U} x_{uv} = 1 \quad \forall v \in V \tag{3.2}$$

Constraint 3.2 ensure that all demand points $v \in V$ are served by primary nodes $u \in U$.

$$\sum_{k \in K} \sum_{i \in I} y_{uik} \leq 1 \quad \forall u \in U \tag{3.3}$$

Constraint 3.3 restricts the allocation of only one type and only one role (primary/backup) per node. For instance, if the node 0 ($u = 0$) of type 5 ($i = 5$) is activated as primary node (k=0), the model cannot activate another node $u = 0$.

$$\sum_{v \in V} x_{uv} \tau_v \leq \sum_{i \in I} y_{ui0} \Psi_{ui} \quad \forall u \in U \tag{3.4}$$

$$\sum_{v \in V} x_{uv}\sigma_v \leq \sum_{i \in I} y_{ui0}\Phi_{ui} \quad \forall u \in U \tag{3.5}$$

Constraints 3.4 and 3.5 ensures that the CPU and RAM capacities of an FN of type i assigned as primary should be greater than or equal to the capacities requested by a demand point assigned to it.

$$1 - (1 - r_u)(1 - r_w) \tag{3.6}$$

$$1 - (1 - \sum_{i \in I} y_{ui0}r_{ui})(1 - \sum_{i \in I} y_{wi1}r_{wi}) \geq R_v x_{uv} \quad \forall u \in U, w \in U, v \in V \tag{3.7}$$

$$1 - (1 - \sum_{i \in I} y_{ui0}r_{ui})(1 - \sum_{i \in I} y_{wi1}z_{uw}r_{wi}) \geq R_v x_{uv} z_{uw} \quad \forall u \in U, w \in U, v \in V \tag{3.8}$$

Constraint 3.8 follows from the reliability of a redundant system presented in Section 2. Recall a redundant system remains on standby a layer of backup devices to guarantee regular operation. Equation 3.6 illustrates the total reliability of a redundant system with two elements in parallel: u and w (*i.e.,* primary and secondary FNs). Equation 3.7 instances the Equation 3.6 with the decision variables into a constraint that assures the overall reliability level achieved with the implemented protection scheme. This level should be greater than or equal to Rv, which is the expected reliability level of a demand point $v \in V$. Equation 3.8 involves the decision variable z_{uw}. The left-hand side of the expression verifies that y_{wi1} is the assigned backup of y_{ui0}; the right-hand side checks that u is the assigned FN attending the demand point v.

The protection schemes $1:1$ and $1:N$ are considered in the FN allocation problem to furnish different protect services in case of FN failure. They are presented in Subsection 3.3.1.

3.3.1 Protection schemes

This section extends the formulation of the FN allocation problem presented in Section 3.3 to furnish a $1:1$ and a $1:N$ shared protection schemes to mitigate the impact of failures of FNs on service provisioning. In the former scheme, if a primary node fails, its demand points are reassigned to a dedicated backup on a different FN of the same type. The secondary node serves as a dedicated backup facility. In the latter scheme, the demand points are reassigned to a secondary FN shared by $N < |U|$ primary nodes if one of them fails. For instance, if the primary FN_{040} of type 4 is attending the demand points 1, 2, and 3, and the primary FN_{110} of type 1 is serving the demand point 0, with $N = 2$, the secondary FN_{241} of type 4 is the shared backup of both primary nodes and will attend the demand points of the first node that fails.

$$\sum_{w \in U} z_{uw} = \sum_{i \in I} y_{ui0} \quad \forall u \in U \tag{3.9}$$

$$\sum_{u \in U} z_{uw} \leq N \quad \forall w \in U \tag{3.10}$$

Constraint 3.9 indicates that a backup node is assigned for each primary active FN. Constraint 3.10 indicates that a backup FN can replace $N \leq |U|$ potential primary nodes. With this constraint, the dedicated protection scheme is a particular case when $N = 1$.

$$z_{uw} \sum_{i \in I} y_{ui0} \Psi_{ui} \leq z_{uw} \sum_{i \in I} y_{wi1} \Psi_{wi} \quad \forall u \in U w \in U \tag{3.11}$$

$$z_{uw} \sum_{i \in I} y_{ui0} \Phi_{ui} \leq z_{uw} \sum_{i \in I} y_{wi1} \Phi_{wi} \quad \forall u \in U w \in U \tag{3.12}$$

Constraints 3.11 and 3.12 ensures that the CPU and RAM capacities of a secondary FN $w \in U$ of type i must be greater than or equal to the capacities of the primary

FN $u \in U$ assigned to it.

3.3.2 Linearization

The linearization of Equation 3.8 was required to simplify the implementation of the reliability constraint. For linearizing the multiplication between the binary decision variables y_{wi1} and z_{uw}, an auxiliary variable aux_{uiw} was necessary:

- $aux_{uiw} \in \{0, 1\}$- the value 1 indicates that the secondary node $w \in U$ assigned as $k = 1$ is active and is backing up the FN $u \in U$ of type i.

Equations 3.13 and 3.14 are necessary constraints regarding this operation.

$$2 \times aux_{uiw} \leq y_{wi1} + z_{uw} \quad \forall u \in U, i \in I, w \in U \tag{3.13}$$

$$y_{wi1} + z_{uw} - 1 \leq aux_{uiw} \quad \forall u \in U, i \in I, w \in U \tag{3.14}$$

$$1 - (1 - \sum_{i \in I} y_{ui0} r_{ui})(1 - \sum_{i \in I} aux_{uiw} r_{wi}) \geq R_v x_{uv} z_{uw} \quad \forall u \in U, w \in U, v \in V \tag{3.15}$$

Constraint 3.15 replaces Equation 3.8 in the model. The left-hand side of expression checks if the reliability rate of the primary nodes ($k = 0$) and corresponding secondary nodes ($k = 1$) protection schemes are enough to meet the reliability demanded. The right-hand side of the expression verifies that the reliability of the demand point v is attended by the actives primary and secondary node u and w.

For the same purpose, we also linearize the multiplication between the decision variables y_{uik} and z_{uw} in Equations 3.11 and 3.12. Taking into account that all the decision variables are binaries, we can describe Equation 3.11 as Equation 3.16

where the left-hand side of the bracket is the difference of CPU capacities between the backup and primary node.

$$\sum_{i \in I} y_{wi1} \Psi_{wi} - \sum_{i \in I} y_{ui0} \Psi_{ui} \quad \begin{cases} \geq 0 & \text{if } z_{uw} = 1 \\ \text{any value} & \text{if } z_{uw} = 0 \end{cases} \tag{3.16}$$

The lower possible value of this difference is $-\Psi_{max}$ where Ψ_{max} is the highest possible CPU capacity in the model. Since Ψ_{max} is a known value, then:

$$\sum_{i \in I} y_{wi1} \Psi_{wi} - \sum_{i \in I} y_{ui0} \Psi_{ui} \quad \begin{cases} \geq 0 & \text{if } z_{uw} = 1 \\ \geq -\Psi_{max} & \text{if } z_{uw} = 0 \end{cases} \tag{3.17}$$

Meaning that:

$$\sum_{i \in I} y_{wi1} \Psi_{wi} - \sum_{i \in I} y_{ui0} \Psi_{ui} \geq -(1 - z_{uw}) \Psi_{max} \tag{3.18}$$

$$\sum_{i \in I} y_{wi1} \Psi_{wi} - \sum_{i \in I} y_{ui0} \Psi_{ui} \geq (z_{uw} - 1) \Psi_{max} \tag{3.19}$$

Thus, Equation 3.11 is updated by Equation 3.20. Similarly, Equation 3.12 is updated by Equation 3.21.

$$\sum_{i \in I} y_{wi1} \Psi_{wi} - \sum_{i \in I} y_{ui0} \Psi_{ui} \geq \Psi_{max} z_{uw} - \Psi_{max} \quad \forall u \in U, w \in U \tag{3.20}$$

$$\sum_{i \in I} y_{wi1} \Phi_{wi} - \sum_{i \in I} y_{ui0} \Phi_{ui} \geq \Phi_{max} z_{uw} - \Phi_{max} \quad \forall u \in U, w \in U \tag{3.21}$$

Constraints 3.20 and 3.21 state that the CPU and RAM capacities of a backup node $w \in U$ are at least the same as its primary node. The values Ψ_{max} and Φ_{max}

represent the largest CPU and RAM capacities, respectively, the CPU and RAM of a type 2 FN.

3.4 Performance Evaluation

This chapter exposes the performance evaluation of the optimization problem detailing the Smart Farms scenarios, experiment setup, and the analysis of the results.

3.4.1 Experiment Setup

This section presents the results obtained for different types of demand points as input to the optimization problem. Figure 3.4 illustrates several scenarios such as coffee, banana, avocado and tomatoes smart farms. Each farm's supply chain has stages with specifics needs and hence, capabilities requirements according to the SFFCs to execute. For instance, production forecasting requires more CPU and RAM capabilities than the ground temperature monitoring of crops; the continued control of pests in crops demands more resources than handling irrigation systems.

Therefore, we classify four types of demand points to analyze the performance of the optimization model considering several farming scenarios. Table 3.3 presents the parameters that define the CPU and RAM capabilities per type of demand. We differentiate low, medium, and high demand points, plus a category that includes all classes. Demands of CPU and RAM are uniformly distributed in the ranges presented in Table 3.3. The designed network infrastructure is composed of 30 to 60 potential FNs nodes and 30 demand points in all cases (*i.e.*, $V = 30$).

Table 3.3: Types of demands

Demand	CPU (MIPS)	RAM (GB)	Potential FNs (U)
Low	1-1240	0.01-0.5	30
Medium	1240-2460	0.5-1	50
High	2460-3680	1-2	60
All	1-3680	0.01-2	40

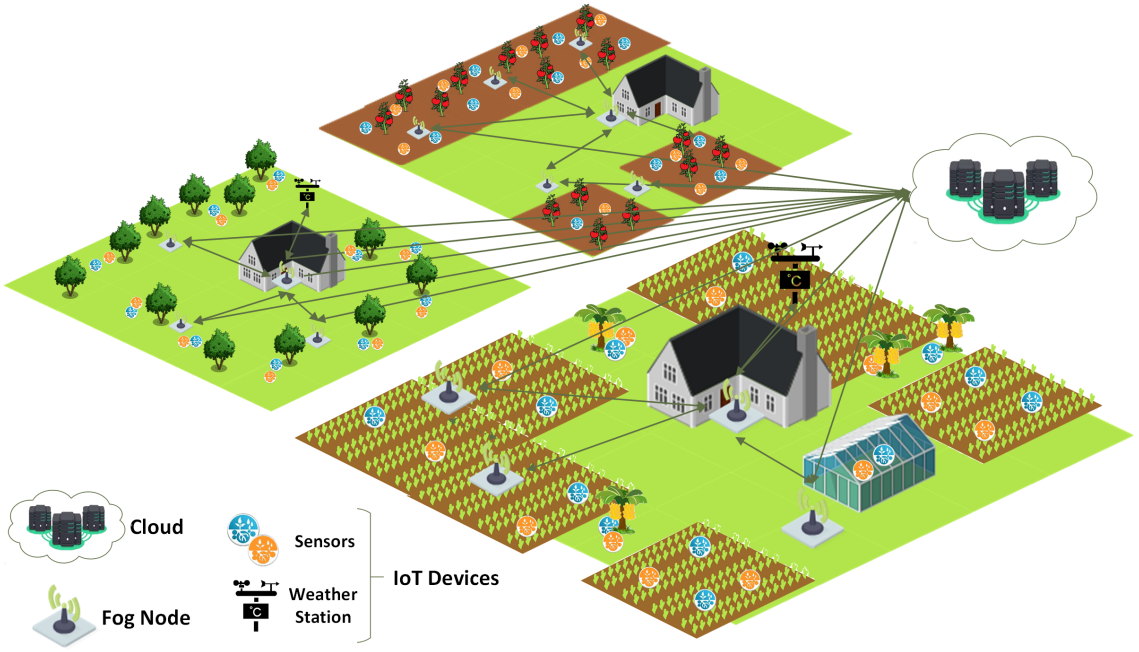


Figure 3.4: Fog Computing-based Smart Farming

Furthermore, we distinguish three types of FNs based on devices commonly instanced in an SF deployment such as Raspberry Pi (zero, 3B+, and 4B models). Table 3.4 sums up the description of each node type classified by CPU and RAM capabilities, reliability rate, and price.

Table 3.4: Node type description

Type	CPU (MIPS)	RAM (GB)	Reliability	Price (USD)
0	1240	0.5	0.50	20
1	2460	1	0.60	45
2	3680	2	0.70	70

The problem formulation presented in Section 3.3 was coded and implemented by Gurobi Optimizer solver version 9.1.1 [130]. Gurobi is a mathematical solver for Linear Programming (LP), Quadratic Programming (QP), and mixed programming problems. We use *gurobipy*, a software library implementation of Gurobi using Python 3.9.0 [131]. Most actions in the Gurobi Python interface are performed by calling methods on Gurobi objects. The most commonly used object is the *Model*.

The proposed model consists of a set of decision variables (Appendix A, lines 98-102), a linear objective function on these variables (Appendix A, line 108), and a set of constraints on these variables (Appendix A, lines 110-144). Once the model is built, the *Model.optimize* function computes a solution (Appendix A, line 147). By default, *optimize* use the concurrent optimizer to solve LP models, the barrier algorithm to solve QP models with convex objectives, and quadratically constrained programming models with convex constraints branch-and-cut algorithm otherwise. The solution is stored in a set of attributes of the model, which can be subsequently queried [132].

3.4.2 Results and Analysis

This section presents the numerical results obtained by the linear programming model presented in this chapter. We analyze two metrics: cost and number of activated FNs, both presented as a function of the required reliability. These metrics are shown for all demand scenarios (low, medium, high, and all classes) to show how the demands impact the final deployment. This evaluation considered farm scenarios requiring reliability levels from 10% to 90%. The reliability rate indicates the probability that a device operates during a year.

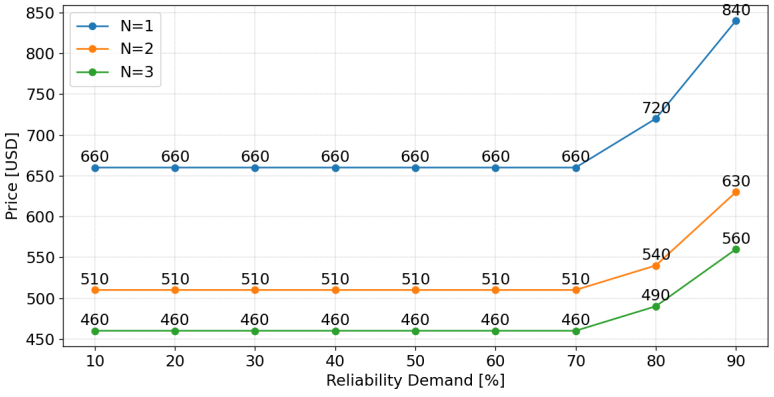


Figure 3.5: Price vs. Reliability - Low demand points

Figures 3.5, 3.6, 3.7, and 3.8 display the results for the cost metric as a function of the required reliability level. $1 : N$ protection schemes were evaluated for N values

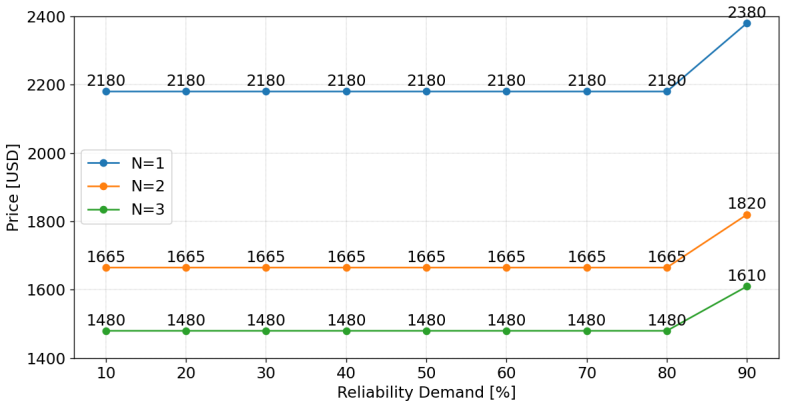


Figure 3.6: Price vs. Reliability - Medium demand points

equal to 1, 2, and 3. The impact of N can be seen for all demand scenarios: for greater values of N, fewer physical resources are required due to more shared backup nodes, reducing the deployment price. Therefore, the dedicated protection scheme is the most expensive. If N increases, the price reduces, but less primary FNs will be covered by backup nodes.

Results in Figures 3.5, 3.6, 3.7, and 3.8 also evince the effect of the required reliability on the deployment. For low demand points (Figure 3.5), the price is constant for $N \leq 70\%$ meaning that the cheapest deployment can guarantee at least 70% reliability. Recall that the cheapest FN equipment has a reliability rate of 50% (see Table 3.4); if both primary and backup nodes are of type 0, achieving redundancy reliability of 0.75 is possible, enough to meet the reliability demands from 10% to 70%. For medium and all demand points (Figure 3.6 and 3.8), this threshold is higher (80%) due to the higher demands of SFFCs that require better equipment, even under lower reliability requirements. Likewise, under the high demands (Figure 3.7), high CPU and RAM capabilities are needed, requiring powerful hardware that already provides high reliability. Consequently, the cost does not vary in function of the reliability demanded.

The ratio between the costs required for 10% and 90% reliability is quite different in the analyzed scenarios. For low demand points (Figure 3.5), a cost 21% to 28% higher is required to achieve 90% reliability, but such increase is only about 9% for

medium demand points (Figure 3.6), and less than 3% for all demand points (Figure 3.8). This trend shows that high reliability has a much more significant effect on scenarios with lower demands, given that more expensive equipment to deal with high demands is usually already designed to provide better reliability.

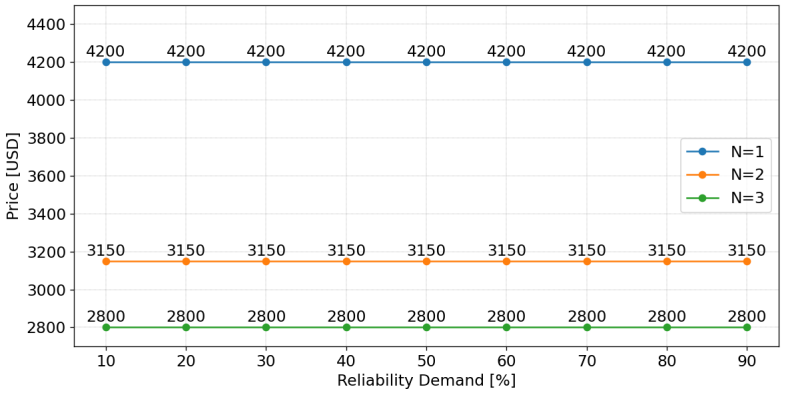


Figure 3.7: Price vs. Reliability - High types of demand points

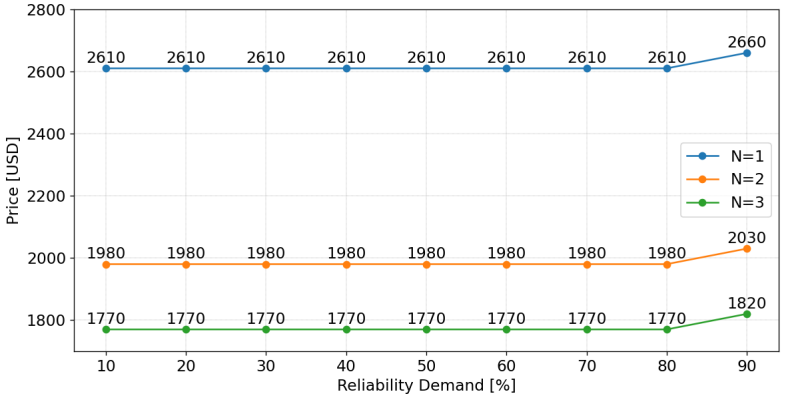


Figure 3.8: Price vs. Reliability - All types of demand points

The evaluation results of heterogeneity in FNs are presented in Figures 3.9, 3.10, 3.11, and 3.12 depicting the total number of activated FNs for different demands. The graphics disclose the activated primary and secondary node type in all the protection schemes. Results were clustered with the same distribution per reliability rate to improve the readability. This clustering corresponds with the price graphs: when the price is constant over the reliability rates, the model finds the same optimal

distribution of the FNs. Different colors indicate the node type as well as its role (primary or backup). Each cluster in the figures has results for different protection scheme levels (N).

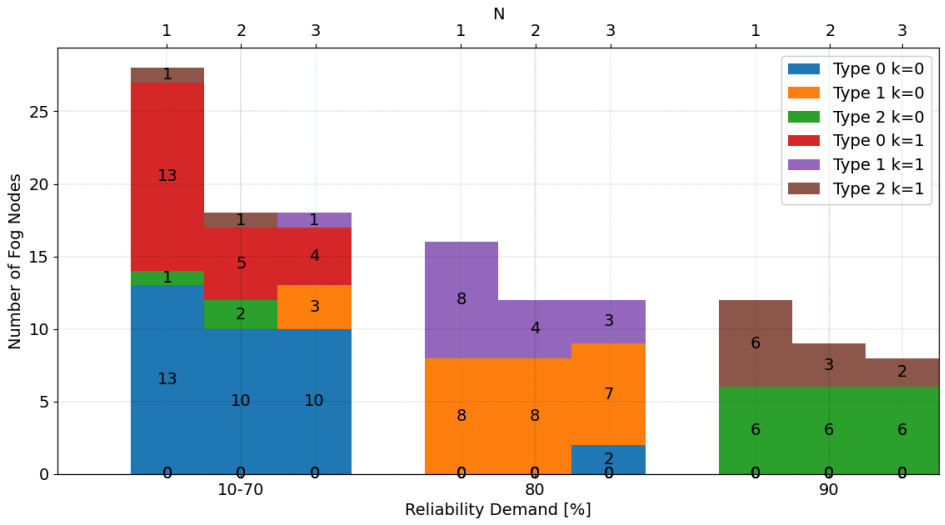


Figure 3.9: Number of activated FNs vs. Reliability - Low demand points

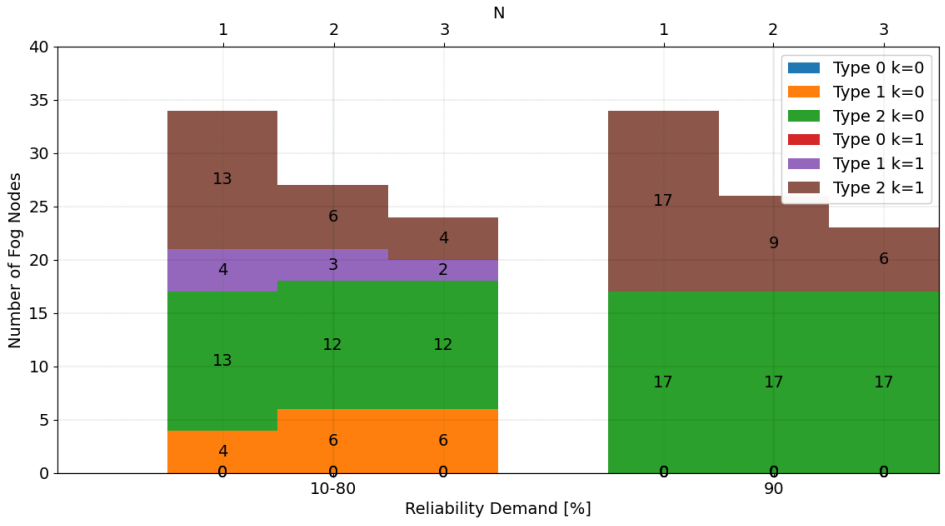


Figure 3.10: Number of activated FNs vs. Reliability - Medium demand points

Three factors affect the number of activated nodes: the demand, value of N, and required reliability. First, higher demands require better resources, but they can also

activate more nodes to serve multiple farms. Therefore, low demands (Figure 3.9) require less than 30 FNs, while high demands (Figure 3.11) can require up to 60 active nodes. Second, higher values of N allow more sharing of resources, reducing the number of hardware equipment. For example, under high demands (Figure 3.11), 30 primary nodes are employed regardless of the required reliability. However, for $N = 1$, 30 backup nodes are activated, one for each primary node, but only 10 nodes are needed for $N = 3$. Finally, higher values of reliability tend to activate fewer nodes, which is more evident for low demands (Figure 3.9). The variations of capacity and price justify these rather contradictory results. When reliability increases, better hardware is needed, raising the costs with infrastructure. However, more expensive hardware offers more CPU and memory capabilities, reducing the number of devices yet increasing costs. The type of FNs also varied for different deployments. The most basic FN hardware, type 0, was only employed in scenarios with low demands (Figure 3.9), which shows that FN deployments with high computational and reliability demands rely on more powerful and expensive hardware. Moreover, the backup nodes usually have the same type as their corresponding primary node.

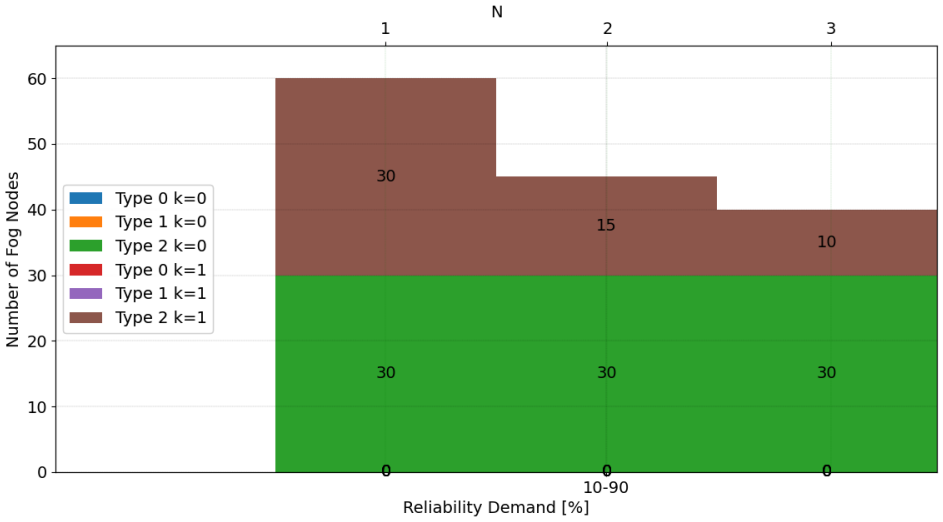


Figure 3.11: Number of activated FNs vs. Reliability - High demand points

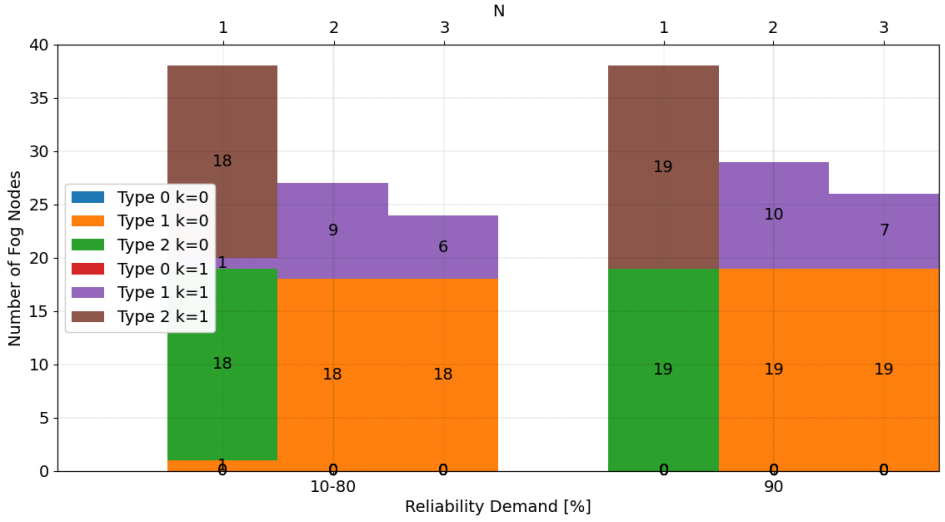


Figure 3.12: Number of activated FNs vs. Reliability - All types of demand points

3.5 Final Remarks

This chapter has addressed the problem of protecting Smart Farms by redundancy techniques where the system stands a set of backups if primary nodes fail. The problem was modeled as an allocation of FNs resources to minimize the cost of implementing a Fog-based SF under reliability constraints. The model performance was evaluated in several smart farms scenarios with low, medium, and high demands and considering parameters of devices commonly instanced in SF deployment such as Raspberry Pi (zero, 3B+, and 4B models). Moreover, we address two redundancy schemes to provide reliability to IoT-based SF services: *i)* $1:1$, a dedicated backup node is reserved for each service demand, and *ii)* $1:N$, a backup node is shared among N potential primary nodes.

The main findings of the numerical results produced by the linear programming formulation proposed in this chapter are summarized as follows. First, protection schemes in FN-based smart farms are possible, guaranteeing high reliability to improve the operation of farms. Second, if backup nodes are shared, the cost associated with the FN infrastructure can be significantly reduced, yet providing good reliability. Third, reliability plays an important role, but it has a more significant impact

on high-reliability demands. Finally, the type of demand has a notable impact on the deployment, significantly increasing required resources and the cost.

Concerning the heterogeneity in FNs, it is essential to mention that its impact is more notable in scenarios with low demands. The above means that for large farms where the services demands are usually high, the more powerful hardware covers the computational and reliability needs at once. Nevertheless, it could be cheaper for small farms with low-demand tasks to deploy low-resource hardware when the reliability demanded is less than 70%.

Chapter 4

A Fog Computing-based Data Reliability Approach in IoT

4.1 Introduction

The IoT has emerged as a suitable technology to collect and transmit data in different domains [133] ranging from agriculture [134], cities [135], and transportation [136] to smart-homes [137]. Reliability is necessary for data collection to guarantee the effectiveness of IoT-based services. Making decisions based on inaccurate data can negatively impact the quality of crops and, consequently, lead to losing money. Several failures affect IoT data collection: *(i)* random errors by lack of sensor reading repeatedly (e.g., scatter in the measured data), *(ii)* spurious reading (i.e., non-systematic reading errors) by a fake measure when some sporadic physical events happen (e.g., if a camera flash triggers when measuring light intensity); and *(iii)* systematic errors such as calibration, loading errors, and environmental errors [16]. These failures cause anomalies or outliers, avoiding reliability in data collection and the good decision-making of end-users (e.g., farmers).

In SF, not all data go to the Cloud, nor do all applications operate in a Cloud style. Note that in developing countries the Internet connectivity is still constrained. Some data and applications must operate in the Fog Tier. For instance, SF applications

can process data or deliver meaningful information to trigger rain and temperature alarms for crops that are too sensitive to climate variations. Thus, SF applications need to work with reliable data in the Fog Tier. These applications would not be useful to the farmer if they operate with unreliable data since it would cause the farming stakeholders to make wrong decisions that, in turn, would affect the yield and control of crops.

In the literature, the works in [27, 32, 28, 30, 114, 115, 116, 117] considered the reliability in IoT and IoT-based SF applications, but they did not exploit the capabilities provided by FC for accomplishing data reliability. FC provides computation, storage, and networking resources closer to end-devices, which allows facing the limitations of IoT applications supported in the Cloud Computing [80, 81]. In developing countries, the SF applications that operate directly at Cloud suffer Internet constraints, outages, or connectivity issues. Such issues will not be present if a Fog Tier is used. The works in [7, 29, 31, 33, 108, 118, 119, 120, 121] managed reliability in FC-based IoT by using optimization, self-adaptability, and redundancy techniques; these investigations did not include mechanisms for outliers detection and data inference for reliability purposes. The works in [122, 123, 124] introduced methods to detect anomalies in datasets without considering an SF environment based on FC and IoT. The work in [125] and [126] proposed mechanisms for detecting and handling outliers by interpolation; this work did not consider FC-and-IoT-based SF.

This chapter proposes an approach intended to provide data collection reliability in Things-based SF, which focuses on outlier detection and treatment. The proposal includes a conceptual Things-Fog-Cloud based architecture that incorporates mechanisms for detecting and treating outliers. The Failure Detection Mechanism (FDM) finds outliers in datasets by ML algorithms. The Failure Recovery Mechanism (FRM) replaces the identified outliers with data inferred by using interpolation techniques. This approach is novel because there is no approach based on ML and interpolation techniques, to the best of our knowledge, aimed to provide data reliability to Things-Fog-Cloud-based SF applications that support decision-making to farming stakeholders. The approach was evaluated by deploying the three Tiers-based architecture in a Colombian coffee smart farm. We run the FDM and FRM mechanisms at the Fog Tier over this real implementation to perform the data reli-

ability testing. The datasets used in the case study contain real information about the coffee crop temperature in different time scales (hour, day, and month) and, further, information about the data collection sensor technologies (Intel, Omicron, and Libelium). Results show our mechanisms achieve high Accuracy, Precision, and Recall as well as low False Alarm Rate (FAR) and Root Mean Squared Error (RMSE) when detecting and replacing outliers with inferred data. Considering the obtained results, we conclude that our approach provides reliable data collection in Smart Farms to support correct decision-making.

The key contributions presented in this chapter are:

- An Things-Fog-Cloud architecture that combines ML and Interpolation techniques to intelligently and automatically provide data reliability on SF applications.
- An ML-based mechanism for outlier detection in IoT-based SF data collection.
- An interpolation-based mechanism for replacing the outliers detected with inferred data.

4.2 Motivation

Let us consider a Smart Coffee Farm as a scenario involving various IoT devices, such as network-connected weather stations and wireless sensors. These IoT devices collect significant and heterogeneous data about the environment and the coffee crop. This data is the primary input for IoT-based applications like data analysis to predict coffee production, traceability to track the coffee source, alarm to inform about variables out of normal ranges, and irrigation systems [25, 26]. Since a coffee farmer makes decisions based on the information that applications provide, data supporting them must be reliable. For instance, if the weather variables' monitoring system fails due to information loss during the data collection, the coffee production estimation may be inaccurate. This problem may affect farmers' annual schedules, avoiding a suitable organization of resources, storage space, and recruitment. Also,

if a coffee quality variable gets outside the range of the standard parameters, such as pH levels at the fermentation phase, the coffee may not achieve exportation quality. This issue may cause the farmer to lose much money (approximately USD 1000 per hectare). Therefore, the data collection must be reliable, aiming to support the smart coffee farm proper operation.

Considering the above scenario, we argue that the Things-Fog-Cloud continuum requires new ways to ensure reliable delivery of data retrieved by IoT-based devices for guaranteeing the proper operation of applications located at the Fog and Cloud tiers. In this sense, the approach proposed in this master thesis is more full-fledged than the ones presented in Section 2.2.2 because it includes a reliability-oriented and Fog-based architecture that incorporates mechanisms for detecting outliers and inferring data to recovering from them in the data collection process.

4.3 Reliable Fog Computing-based Architecture

A Fog hierarchical architecture includes three tiers called Cloud, Fog, and Things [89]. The Cloud Tier comprises one or more Data Centers for providing services that consume many computational resources (e.g., big data analysis). The Fog Tier includes Fog Nodes (FNs). An FN is any physical or virtual entity (*e.g.,* routers, switches, wireless access points, video surveillance cameras, controllers, and servers) that improves the interaction between the Things Tier and Cloud Tier to enhance IoT-based services, such as data collection based on wireless sensors, tracking systems, and, in general, delay-sensitive applications. The Things Tier consists of IoT-enabled devices including sensor nodes. This Tier is responsible for interacting with the environment and sending data to the Fog Tier [95].

Figure 4.1 presents our Fog-based and reliability-oriented architecture. Unlike the traditional Fog-based architectures [8, 83], our architecture separates the Fog Tier into two layers: categorizing resources in layers depending on domain requirements leads to improve IoT reliability by properly locating management services [138]. As making decisions with reliable data is pivotal for IoT-based applications, we locate FDM and FRM mechanisms in Layer 2 FN because it is closest to the IoT devices. In

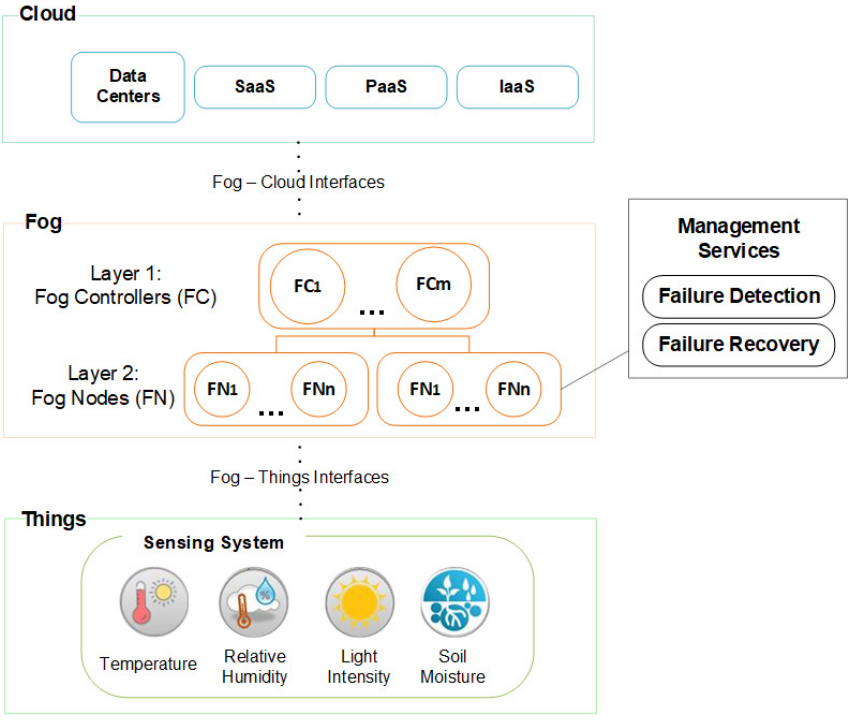


Figure 4.1: Layer-based Fog Hierarchical Architecture

this way, we avoid applications running on upper Tiers operate with inaccurate data. Note that outliers in collected data can negatively impact statistical analysis and the accuracy in estimation and forecasting models of harvest, leading to making wrong decisions and, consequently, lost money. Then, Layer 2 FN comprises several edge computing nodes that collect data from IoT devices and host our reliability-targeted mechanisms. FDM is to detect outliers (see Section 4.4), while FRM is to infer correct values for replacing the detected outliers (see Section 4.5). Layer 1 includes Fog Controllers (FCs) to coordinate the network tiers and perform preliminary data analytic because they have more processing, storage, and networking resources than Layer 2 FN.

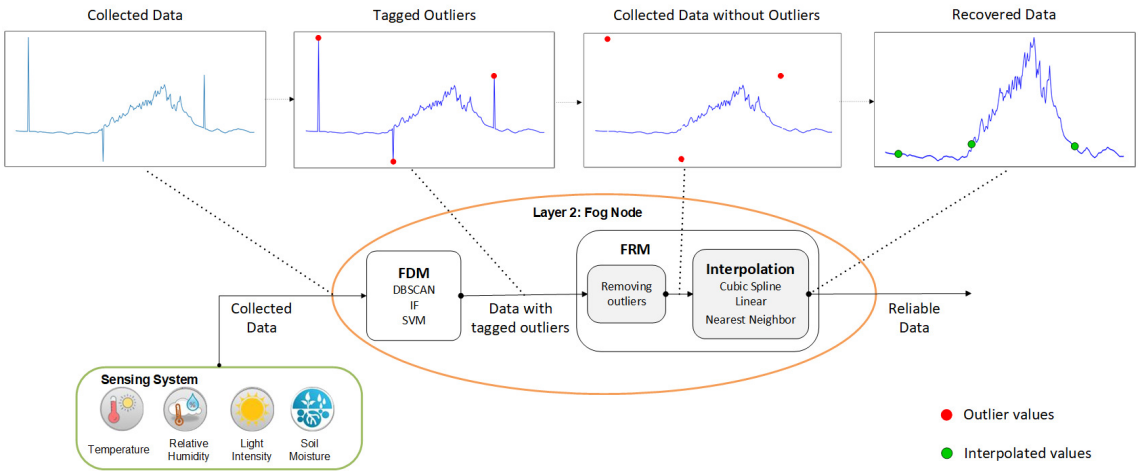


Figure 4.2: Data cleaning mechanisms to provide reliability

4.4 Failure Detection Mechanism

FDM aims at identifying and isolating outliers from correct data. These outliers can happen by failures, such as a sensor with a damaged battery and incorrect reading. The main benefits of detecting outliers in the Fog Tier are: characterizing normal and abnormal data for early treatment of outliers and ensuring data quality before using it in data analytic and forecasting applications. FDM receives data from sensors and identifies the data's failures as outliers (*i.e,* an abnormal and extreme observation of data) using ML algorithms. FDM tags the outliers at the dataset. The dataset with these tagged outliers is the input to FRM.

FDM considers three well-known outlier detection techniques [139]: Clustering-based, Isolation-based, and Classification-based. Clustering-based approaches group similar data instances into clusters with the same behavior [140]. Data instances are identified as outliers if they do not belong to clusters or generate significantly smaller clusters than other ones. The dissimilarity measure between two data instances is the Euclidean distance. Isolation-based approaches focus on separating outliers from the rest of the data points instead of profiling normal ones [141]. Classification-based approaches learn a classification model using the set of data instances (training) and classify an unseen instance into one of the learned (normal/outlier) class (testing) [142].

For the Clustering-based approach, FDM uses the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) due to it is a well-known and established grouping algorithm that does not need the desired number of clusters as K-means does [143]. DBSCAN groups the observations of a dataset into high and low-density clusters using a simplified minimum density level estimation based on distance radius named epsilon (esp), minimum points, and thresholds for the number of neighbors [144]. DBSCAN mainly needs two parameters: min_samples, and esp. With preliminary tests, we verify that varying the parameter min_samples does not affect the outliers detection performed by FDM due to we are considering the higher density cluster, such as the one with the correct values, and the other ones, such as clusters with outliers. The eps parameter controls the maximum distance between two samples. If eps is too small, FDM cannot cluster most data. If eps is too large, FDM can merge all clusters into a single one [145].

For the Isolation-based approach, we use the Isolation Forest (IF) algorithm that poses the ability to identify anomalies (outliers in our approach) from a dataset. This algorithm performs recursive random splits on attribute values generating trees [123] that can isolate any data point from the rest of the data. Random partitioning produces noticeably shorter paths for outliers. Hence, when a forest of random trees collectively produces shorter path lengths for particular samples, they have a high probability of being outliers [146]. For IF, the parameter of contamination defines the proportion of outliers in the dataset [146]. If the contamination is set too low than the real rate of outliers, the model will not detect them. If the parameter is set too high, the number of False Positives will increase.

For the Classification-based technique, we use the Support Vector Machine (SVM) to determine if an instance falling outside a boundary is an outlier [147]. SVM separates the data from different classes by fitting a hyperplane between them, which maximizes the separation [148]. In SVM, the parameter "nu" represents the fraction of outliers in the dataset. This parameter is analogous to the contamination parameter in IF. If nu is set up to a too low value than the actual number of outliers, the model will not detect all outliers. If it is set higher than the actual number of outliers, the model will detect false normal values as outliers.

4.5 Failure Recovery Mechanism

FRM infers data to replace outliers without losing significance in data, aiming to achieve reliability and, consequently, accurate models in IoT-based applications. Figure 4.2 depicts inputs and outputs in FRM. From a high-abstraction level, FRM operates as follows. It receives the dataset with the outliers tagged. Then, it removes and replaces them with data inferred by interpolation techniques. Finally, it delivers the corrected data to applications in the Things-Fog-Cloud continuum for further processing. These applications will be reliable regarding data due to the FDM and FRM operation.

FRM infers data to replace outliers by considering three algorithms: Cubic Spline, Linear, and Nearest Neighbor. Cubic Spline is an interpolation method that returns the straight line connecting data points with a polynomial function to obtain a continuous and smooth curve [149]. Linear interpolation is a curve fitting method using linear polynomials to construct new data points within the range of a discrete set of known data points [150]. Nearest Neighbor interpolation is a proximal interpolation method of multivariate interpolation in one or more dimensions used in image processing [151]. The Nearest Neighbor algorithm takes a rounded value of the expected position and finds the closest data value at the integer position [152].

4.6 Case Study in a Colombian Coffee Smart Farm

This chapter exposes the evaluation of the data reliability approach as follows. Initially, the architecture introduced in Section 4.3 was implemented and deployed in a Colombian Coffee Smart Farm scenario. Then, the temperature datasets at the Colombian Coffee Smart Farm were collected. After, the FDM and FRM at the Fog Tier of the mentioned architecture were executed. Finally, the proposed mechanisms' effectiveness to deliver reliable data were tested regarding Accuracy, Recall, Precision, FAR, F-Score, and RMSE.

4.6.1 Scenario

Figure 4.3 depicts the implemented and deployed architecture in a Colombian Coffee Smart Farm, introduced in Section 4.3. This network aims at providing reliable data for coffee farm services such as watering, fertilizing, harvesting, and forecasting production. The farmers make important decisions to increase their earnings based on the data collected and the information obtained by such services; the quality of historical data, such as temperature, is crucial to their Accuracy. Farmers need to trust in the information offered by services; therefore, it is pivotal to achieve high data reliability.

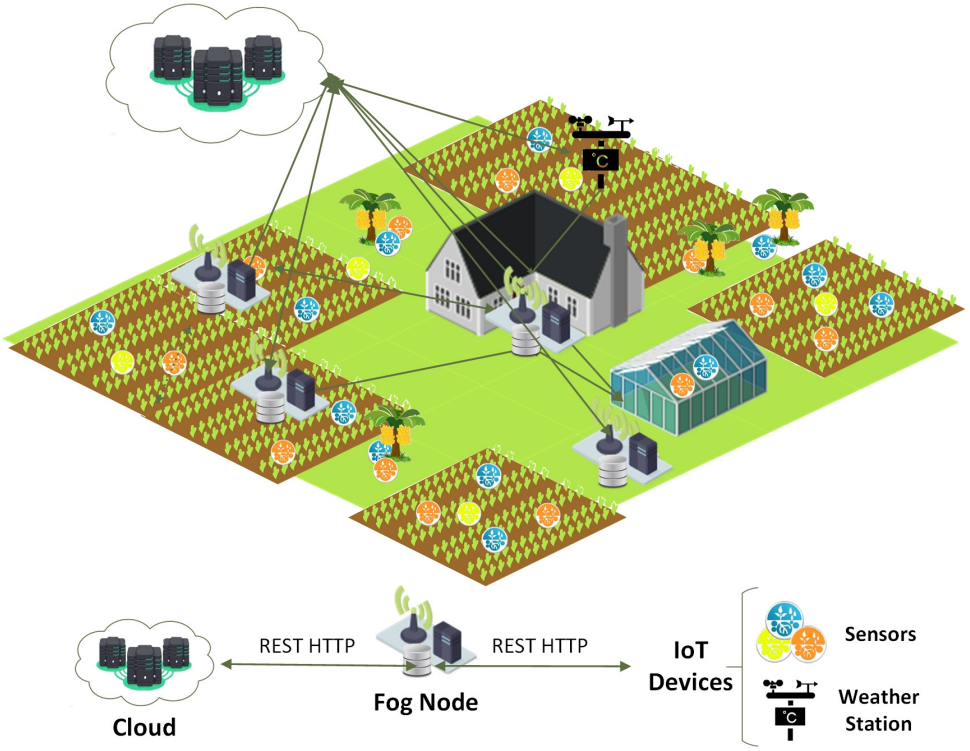


Figure 4.3: Coffee Smart Farming

The Things Tier includes devices such as sensors of temperature, humidity, light, and moisture. The sensors cover three hectares of a coffee crop, the coffee processing center, and the coffee storage center. In particular, this tier comprises: *(i)* a weather station Smart Agriculture PRO from Libelium - Waspmote Plug and Sense! [153] with external solar panels, which supports several radio technologies (ZigBee, Wi-Fi,

4G, SigFox, LoRaWAN) and sensors for monitoring soil moisture, solar radiation, atmospheric pressure, pluviometer, air temperature, and air humidity. This station covers all the three coffee farm hectares. *(ii)* a Datalog X-PRO from OMICRON [154] with an Internal rechargeable battery, which supports Wi-Fi, SigFox and 3G, and sensors for soil moisture, air temperature, and air humidity. This Omicron device operates in just one of the hectares of the coffee farm. *(iii)* sets of new low-cost sensor-tags from INTEL that support BlueTooth Low Energy, ZigBee, and Wi-Fi sensors for temperature and humidity. A set operates in each hectare of the coffee crop. The primary communication protocol used at the Things Tier was IEEE 802.11n (i.e., IEEE 802.11ax, IEEE 802.11 b, and IEEE 802.15.4) because it is the protocol supported by all the IoT devices deployed in the case study farm. Using IEEE 802.11 is not necessary to pay for a scription, such as with SigFox or 3G/4G.

The Fog Tier includes an FN per hectare and a Fog Controller. In this case study, a single Fog Controller and three FNs were used due to the farm extension and the few IoT devices generating data. FN collects data from the Things Tier and runs FDM and FRM for providing reliable data to the Fog Controller; each FN functions on an ESP32 module and a Raspberry Pi 3 Model B with integrated Wi-Fi and Bluetooth connectivity. FNs are separated between 80 and 100 meters (approximately) from each other. The Fog Controller Node runs on a Raspberry Pi 4 Model B that manages and processes the farm information.

The Fog Tier communicates with the others using a REST-based services style by the HTTP protocol; it was necessary to use a standardized north interface that would allow simple interaction with the Cloud. This node sends aggregated information to the Private Cloud Tier that performs an in-depth analysis of the information provided by lower tiers via a Linux VPS server over Hyper-V. This server offers the following farm services: environmental variables monitoring, historic coffee production, production forecast, and IoT infrastructure management.

4.6.2 Coffee Smart Farming Datasets

We conduct evaluation experiments with three temperature datasets (Table 4.1). The first one, named "Per_Hour", contains data collected with the Omicron device every minute and a half from 14:00h to 14:57h on November 5th of 2019. The second dataset named "Per_Day" was generated from an Intel device collecting data every 5 minutes on the January 6th of 2020. The third dataset called "Per_Month" was formed by the Libelium weather station collecting data every two minutes (approximately) from November 10th to December 10th of 2019. For testing purposes, we included randomly 1%, 5%, and 10% of outliers in each of the three datasets above mentioned using the InterQuartile Range (IQR) [155], generating in total nine datasets. Three datasets "Per_Hour" with 24 instances collected by Omicron device with 1%, 5%, and 10% of outliers, respectively. Three datasets "Per_Day" with 288 instances collected by Intel device with 1%, 5%, and 10% of outliers, respectively. Three datasets "Per_Month" with 22532 instances collected by Libelium device with 1%, 5%, and 10% of outliers, respectively. The datasets and codes used in this chapter are located in [156].

Table 4.1: Datasets for coffee Smart Farming

Name	Scale	Number of instances	Technology
Per_Hour	Hour	24	Omicron
Per_Day	Day	288	Intel
Per_Month	Month	22532	Libelium

Table 4.2 presents the structure of datasets mentioned above. The columns represent the features and the row their format. The first two features express the date and time in the human-readable and in Unix timestamp format, respectively. The third feature (value_temp) is the original temperature data in degrees Celsius collected by the Omicron, Intel, or Libelium devices. The fourth feature (Test_1) contains the temperature data, including the outliers randomly created.

Table 4.2: Dataset Structure

date	timestamp	value_temp	Test_1
(dd/mm/yyyy hh:mm:ss x.x.)	(int)	(float)	(float)

4.6.3 Test Environment

We evaluate FDM and FRM in a virtual machine running a Ubuntu 64-bit operative system and using Python version 2.7.17 and R version 3.6.3 for the ML and Interpolation algorithms, respectively. In Python, we deploy a ML library by using scikit-learn [157]; in particular, *DBSCAN* from *sklearn.cluster*, *IsolationForest* from *sklearn.ensemble*, and *OneClassSVM* from *sklearn.svm*. In R [158], we deploy the following functions to interpolate: *spline* for Cubic Spline interpolation, *approx* for Linear interpolation, and *loess.smooth* for Nearest Neighbor interpolation. It is worth mentioning that we use the CRISP-DM methodology for the construction of the mechanisms [159]. This methodology consists of the following steps:

1. Business understanding focuses on understanding the objectives and requirements from a business perspective (i.e., provide reliable data in SF).
2. Data understanding takes care of the initial data collection and allows becoming familiar with the data (i.e., extract the datasets from the architecture deployed in our case study farm).
3. Data preparation covers all the activities necessary to build the final dataset.
4. During modeling, apply data mining techniques to our data, including the tunning of their parameters to achieve the best results (i.e., FDM and FRM construction).
5. The model's evaluation to determine if they are useful to the business needs.
6. Deployment involves exploiting the models within a production environment (i.e., deployment of FDM and FRM in the Colombian coffee farm).

4.6.4 Performance Metrics

We evaluate FDM's ability to identify outliers by using the metrics involved in the confusion matrix: Accuracy, Recall, Precision, FAR, and F-Score [160]. We use the classical metrics used by other works in the literature for evaluating ML algorithms and Interpolation techniques [161, 124, 139, 162]. Table 4.3 presents the confusion matrix, the terms "positive" and "negative" refer to the classifier's prediction (*i.e.,* normal or outlier). The terms "true" and "false" refer to whether the prediction corresponds to the proper observation. The True Negative indicates the outliers detected correctly. False Positive denotes values classified as normal that were outliers. False Negative exposes outliers incorrectly identified. True Positive states the well-classified normal values.

Accuracy is the proportion of normal and outlier values correctly classified among the total number of classifications (see Equation 4.1). Accuracy answers the question: how many classifications the algorithm identified correctly?. Recall refers to the percentage of total normal values classified correctly by the algorithm (see Equation 4.2). Recall answers the question: how many instances the algorithm identified correctly?. Precision is the fraction of normal values that are properly-identified among the instances classified as normal (see Equation 4.3). Precision answers the question: how many instances the algorithm predicted correctly?. F-Score is the harmonic mean of Precision and Recall (see Equation 4.4). F-Score is best if there is a balance between Precision and Recall. False Alarm Rate is the percentage of falsely detected normal values of the instances classified as outliers (see Equation 4.5). FAR answers the question: how many outliers the algorithm identified incorrectly?.

Table 4.3: Confusion Matrix

	Predicted outlier	**Predicted normal**
Actual outlier	True Negative (TN)	False Positive (FP)
Actual normal	False Negative (FN)	True Positive (TP)

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4.1}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{4.2}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{4.3}$$

$$F - Score = \frac{(2 \times Recall \times Precision)}{(Recall + Precision)} \tag{4.4}$$

$$FAR = \frac{FP}{(FP + TN)} \tag{4.5}$$

We evaluate FRM's ability to infer data to replace outliers by using RMSE [163]. RMSE is a method of measuring the difference between values predicted by a model and their actual values. RMSE measures the amount of error between any two datasets. In this vein, we calculate the RMSE of the original temperature datasets versus the datasets interpolated to identify the most accurate technique. The less RMSE, the better the performance of the interpolation technique.

4.6.5 Failure Detection Evaluation

We conducted experiments with different datasets to evaluate, in terms of Accuracy, Recall, Precision, FAR, and F-Score, several candidate algorithms for realizing FDM; this mechanism tags the detected outliers and generates the tagged outliers dataset that is the FRM's input. This evaluation aims at selecting an algorithm for carrying out FDM. In this sense, we define the best performance condition for outliers detection algorithm: high Accuracy, high Precision, and low FAR. In particular, we vary the parameters from each algorithm described in Section 4.4 as follows. For DBSCAN, the eps (e) parameter from 0.1 to 0.8 (with min_samples set in 3). For IF, the contamination (c) parameter from 0 to 0.5. For SVM, the nu parameter from 0.01 to 0.1. We evaluated the algorithms using the datasets described in Table 4.1 aggregating to each of them 1%, 5%, and 10% of outliers.

Table 4.4: Comparison Per_Hour dataset

		Accuracy	Recall	Precision	F-Score	FAR
1% Outliers	DBSCAN (e=0.3 - 0.8)	100	100	100	100	0
	IF (c=0 - 0.04)	100	100	100	100	0
	SVM (nu=0.01 - 0.1)	83.33	82.61	100	90.48	0
5% Outliers	DBSCAN (e=0.3 - 0.8)	100	100	100	100	0
	IF (c=0.05 - 0.08)	100	100	100	100	0
	SVM (nu=0.09 - 0.1)	100	100	100	100	0
10% Outliers	DBSCAN (e=0.2)	100	100	100	100	0
	IF (c=0.09 - 0.1)	100	100	100	100	0
	SVM (nu=0.09 - 0.1)	79.16	80.95	94.44	87.18	33.33

Table 4.4 compares in the Per_Hour dataset the candidate algorithms for performing FDM. DBSCAN and IF obtained the same high performance in any outliers' percentage, meaning these algorithms can operate correctly with a small dataset and few outliers. SVM obtained the worst performance; in this dataset with 10% of outliers, this algorithm got 33% and 80% of FAR and Recall, respectively, which indicates SVM did not find all the outliers and present many false positives. These results are because SVM ignores the spatial correlation of neighboring nodes, which makes the results of local outliers inaccurate [139].

Table 4.5: Comparison Per_Day dataset

		Accuracy	Recall	Precision	F-Score	FAR
1% Outliers	DBSCAN (e=0.2 - 0.6)	100	100	100	100	0
	IF (c=0.01)	100	100	100	100	0
	SVM (nu=0.01)	100	100	100	100	0
5% Outliers	DBSCAN (e=0.2 - 0.6)	99.65	100	99.64	99.82	7.14
	IF (c=0.07)	97.56	97.45	100	98.71	0
	SVM (nu=0.09)	95.83	95.62	100	97.76	0
10% Outliers	DBSCAN (e=0.1 - 0.2)	99.65	100	99.62	99.81	3.45
	IF (c=0.2)	89.93	88.80	100	94.07	0
	SVM (nu=0.1)	98.96	99.61	99.23	99.24	6.90

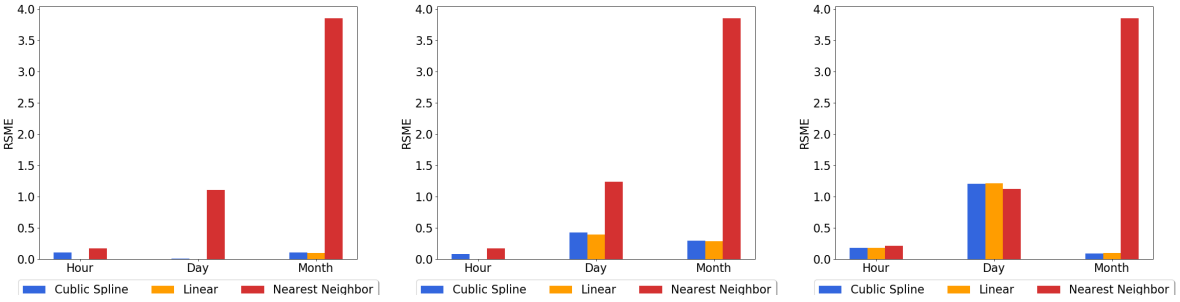
Table 4.5 compares in the Per_Day dataset the candidate algorithms for carrying out FDM. All evaluated algorithms' performance decreased in this dataset (12 times bigger than the Per_Hour dataset). The three evaluated algorithms obtained their

best performance in this dataset with 1% of outliers, confirming they function well when operating with a small dataset and few outliers. Overall, DBSCAN obtained the best results for the evaluated metrics. However, note that this algorithm got 7% and 3.5% of FAR when operating with 5% and 10% of outliers in the dataset, respectively, meaning in the datasets mentioned, the outliers are closer to standard data (DBSCAN has problems to identify this type of outliers). SVM obtained excellent results in almost all evaluated metrics but slightly lower than DBSCAN did. Also, FAR in SVM increased up to 6.9 when the dataset contains 10% of outliers. The IF algorithm got excellent results regarding Precision, F-Score, and FAR. However, its Accuracy and Recall are around 97% and 90% when this dataset contains 5% and 10% of outliers, respectively. This algorithm had problems with false positives.

Table 4.6: Comparison Per_Month dataset

		Accuracy	**Recall**	**Precision**	**F-Score**	**FAR**
1% Outliers	DBSCAN (e=0.1 - 0.2)	99.99	100	99.99	99.99	1.33
	IF (c=0.02)	98.98	98.97	100	99.48	0
	SVM (nu=0.02)	99.02	99.01	100	99.50	0
5% Outliers	DBSCAN (e=0.1)	99.78	100	99.77	99.89	0.44
	IF (c=0.06)	98.90	98.94	100	99.42	0
	SVM (nu=0.07)	97.89	97.78	100	98.88	0
10% Outliers	DBSCAN (e=0.1)	99.46	100	99.41	99.70	5.6
	IF (c=0.1)	99.57	99.52	100	99.76	0
	SVM (nu=0.1)	98.23	98.80	99.25	99.02	7.09

Table 4.6 compares in the Per_Month dataset the candidate algorithms for performing FDM. All evaluated algorithms obtained excellent results regarding Accuracy, Recall, Precision, and F-Score. Accuracy and Precision in DBSCAN decreased slighter to 99.7% and 99.4% when the percentage of outliers moved from 5% and 10%, respectively. In this dataset, DBSCAN also had problems with false positives (FAR=5.6 with 10% of outliers), indicating limitations to identify outliers closer to normal data. Regarding SVM, it is to highlight that its false positives increase when the percentage of outliers is equal to 10% (FAR=7.09), which suggests inefficiency in classifying many outliers. FAR obtained by IF is perfect because the contamination parameter used was very close to the real percentage of outliers existing in the dataset; this is difficult to know in practice for an online operation.



(a) 1% Outliers (b) 5% Outliers (c) 10% Outliers

Figure 4.4: Comparison Interpolation

Based on the results above, we consider DBSCAN as the best option for carrying out FDM. DBSCAN obtained an excellent performance for marking outliers over all the tests with a FAR lower than 6%, a perfect Recall and an Accuracy, Precision, and F-Score greater than 99% for the most extensive dataset. Furthermore, DBSCAN does not require a re-setting of the parameter eps when the outliers increase. Note that IF got the highest Precision and lowest FAR. However, the IF's contamination parameter needs to be re-set when the outliers raise to keep high performance. The IF's false negatives will increase significantly if the real number of outliers is lower than the contamination. In turn, the false positives of IF will increase when the actual number of outliers is higher than the contamination. This re-setting is unpractical because it is hard to know the percentage of outliers previous to the algorithm's training.

4.6.6 Failure Recovery Evaluation

We conducted experiments with the datasets outputted by FDM for evaluating, regarding RMSE, several interpolation techniques that allow replacing the outliers with 'accurate' data. Figure 4.4 shows the Nearest Neighbor technique obtained the worst RMSE; Nearest Neighbor selects the nearest datum's value without considering neighboring data values, which tends to increase noise, especially in the Per_Month dataset. From a conceptual perspective, the Cubic Spline should offer better results than the Linear [164]. Still, our experimental results indicate a lower RMSE with

the Linear, which is due to the data points in our datasets are closely near (our data samples are measurements of temperature). Thus, the connection between each data point by a straight line allows achieving high-accuracy interpolation.

4.7 Final Remarks

This chapter introduced an FC-based architecture approach that incorporates a mechanism for detecting outliers and another for inferring data intended to replace them. The evaluation demonstrated the approach's effectiveness in a real network deployed in a Colombian Smart Coffee Farm. For the failure detection mechanism, the DBSCAN algorithm is selected due to it presented an excellent performance for marking outliers over all the tests with a FAR lower than 6%, a perfect Recall as well as an Accuracy, Precision, and F-Score greater than 99% for the most extensive dataset. The linear interpolation for the failure recovery mechanism is selected because it infers data with low RMSE allowing replacing the detected outliers properly. Considering the obtained results, we concluded the proposed approach is suitable for providing reliability in the IoT-based Smart Farming data collection process and supports the correct decision-making.

Chapter 5

Conclusions and future work

The research performed in this master thesis was guided by the following research question: **How to provide reliability in the collection and transmission of data in an IoT-based Smart Farming?**

This question was addressed by a mechanism composed of two approaches: system and data reliability. The first approach introduces an allocation optimization model of FNs' resources (processing and memory) to meet the farm users' demands while maintaining the reliability requirements. The model provides overall reliability by redundancy techniques where the system remains, in stand-bay, a set of nodes configured in parallel as backups. Thus, if a primary node fails, the secondary will process all the tasks of the first one giving continuity to the farm services. Moreover, the model includes two redundancy schemes to provide reliability to IoT-based SF services: a dedicated backup node reserved for each service demand and a backup node shared among N potential primary nodes.

In particular, we mathematically modeled the system reliability approach by a linear programming formulation to find the optimal number of activated FNs, minimizing the deployment cost under reliability constraints. We also conducted a deep analysis of FNs heterogeneity's influence in our model and introduced new concepts to model the proper allocation of resources: vSFFs and SFFCs. To evaluate the performance model, two metrics were analyzed: cost and number of activated FNs, both presented

as a function of the required reliability. This approach differentiates low, medium, and high demand points, plus a category that includes all classes. The analysis results disclose:

- If backup nodes are shared, the cost associated with the FN infrastructure can be significantly reduced, yet providing good reliability.
- Reliability has a more significant impact on the deployment costs in high-reliability demands.
- The type of demand has a notable impact on the deployment, significantly increasing required resources and the cost.

The second approach introduces a Things-Fog-Cloud architecture that combines ML and Interpolation techniques to intelligently and automatically provide data reliability on SF applications. This approach provide data reliability by detecting and treating outliers from the IoT data collection by mechanisms located at the Fog Tier. The FDM finds outliers in datasets by ML algorithms. The FRM replaces the identified outliers with data inferred by using interpolation techniques. We consider these approaches a step further in reliable Smart Farms since they provide foundations for developing farmers' decision-making applications dependably.

The FC-based architecture was evaluated in a real network deployed in a Colombian Smart Coffee Farm to demonstrate the efficiency of the mechanisms. The quantitative results of the FDM and FRM evaluation revealed that:

- For the failure detection mechanism, the DBSCAN algorithm presented an excellent performance for marking outliers over all the tests with a FAR lower than 6%, a perfect Recall as well as an Accuracy, Precision, and F-Score greater than 99% for the most extensive dataset.
- For the failure recovery mechanism, the linear interpolation infers data with low RMSE allowing replacing the detected outliers properly.

Considering the obtained results, this master dissertation concluded the proposed approaches are suitable for providing reliability in the IoT-based Smart Farming. This master thesis support the dependably farmer's decision-making application by continuity provide correct service and giving the accurate data on Smart Farms. It is important to highlight that all the approaches are generic, then it can be apply to other IoT-based FC domains.

As future work, the model's multi-objective formulation optimizing energy consumption, latency, and node location are considered, and the model implementation in other domains such as SDN. Techniques or mathematical process to reduce significantly the model execution time is also consider as a future work.

Furthermore, the involvement of more features into datasets (such as humidity, pressure, and light), and the improvement of failure detection by the ability to differentiate outliers from events of interest are also considered.

Bibliography

[1] V. Moysiadis, P. Sarigiannidis, V. Vitsas, and A. Khelifi, "Smart farming in europe," *Computer Science Review*, vol. 39, p. 100345, 2021.

[2] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming–a review," *Agricultural Systems*, vol. 153, pp. 69–80, 2017.

[3] P. P. Chandak and A. J. Agrawal, "Smart farming system using data mining," *International Journal of Applied Engineering Research*, vol. 12, no. 11, pp. 2788–2791, 2017.

[4] Z. Khan, M. Zahid Khan, S. Ali, I. A. Abbasi, H. Ur Rahman, U. Zeb, H. Khattak, and J. Huang, "Internet of things-based smart farming monitoring system for bolting reduction in onion farms," *Scientific Programming*, vol. 2021, 2021.

[5] G. R. Mendez and S. C. Mukhopadhyay, "A wi-fi based smart wireless sensor network for an agricultural environment," in *Wireless Sensor Networks and Ecological Monitoring*. Springer, 2013, pp. 247–268.

[6] M. J. O'Grady and G. M. O'Hare, "Modelling the smart farm," *Information Processing in Agriculture*, vol. 4, no. 3, pp. 179–187, 2017.

[7] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, and A. F. Skarmeta, "Smart farming IoT platform based on edge and cloud computing," *Biosystems Engineering*, vol. 177, no. xxxx, pp. 4–17, 2019.

[8] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Future Internet of Things and Cloud (Fi-Cloud), 2014 International Conference on*. IEEE, 2014, pp. 464–470.

[9] M. Balasubramaniyan and C. Navaneethan, "Applications of internet of things for smart farming–a survey," *Materials Today: Proceedings*, 2021.

[10] M. M. Subashini, 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, pp. 456–468, 2018.

[11] P. 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, 2016.

[12] M. Dalton, "Reliable communication is a key to iot growth," 2018. [Online]. Available: http://www.analog.com/en/technical-articles/reliable-communication-is-a-key-to-iot-growth.html

[13] Z. Bakhshi, G. Rodriguez-Navas, and H. Hansson, "Dependable Fog Computing: A Systematic Literature Review," *Proceedings - 45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2019*, pp. 395–403, 2019.

[14] G. Jesus, A. Casimiro, and A. Oliveira, "A Survey on Data Quality for Dependable Monitoring in Wireless Sensor Networks," *Sensors*, vol. 17, no. 9, p. 2010, 2017.

[15] F. Thiam, M. Mbaye, and A. M. Wyglinski, "Generic reliability analysis model of iot: Agriculture use case," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–5.

[16] H. B. Mitchell, *Multi-sensor data fusion: an introduction*. Springer Science & Business Media, 2007.

[17] D. Ganesan, D. Estrin, and J. Heidemann, "Dimensions: Why do we need a new data handling architecture for sensor networks?" *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 143–148, 2003.

[18] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Sensor Network Protocols and Applications, 2003.*

Proceedings of the First IEEE. 2003 IEEE International Workshop on. IEEE, 2003, pp. 139–148.

[19] H. Kopetz, *Real-time systems: design principles for distributed embedded applications*. Springer Science & Business Media, 2011.

[20] C. M. Coman, G. D'amico, A. V. Coman, and A. Florescu, "Techniques to improve reliability in an iot architecture framework for intelligent products," *IEEE Access*, vol. 9, pp. 56 940–56 954, 2021.

[21] M. A. Mahmood, W. K. G. Seah, and I. Welch, "Reliability in wireless sensor networks: A survey and challenges ahead," *COMPUTER NETWORKS*, vol. 79, pp. 166–187, 2015.

[22] D. Pivoto, P. D. Waquil, E. Talamini, C. Pauletto, S. Finocchio, V. Francisco, D. Corte, and G. D. V. Mores, "Scientific development of smart farming technologies and their application in Brazil," *Information Processing in Agriculture*, vol. 5, no. 1, pp. 21–32, 2018.

[23] R. Divya and R. Chinnaiyan, "Reliability evaluation of wireless sensor networks (rewsn—reliability evaluation of wireless sensor network)," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2017, pp. 847–852.

[24] A. Walter, R. Finger, R. Huber, and N. Buchmann, "Opinion: Smart farming is key to developing sustainable agriculture," *Proceedings of the National Academy of Sciences*, vol. 114, no. 24, pp. 6148–6150, 2017.

[25] D. C. Corrales, J. C. Corrales, and A. Figueroa-Casas, "Towards detecting crop diseases and pest by supervised learning," *Ingeniería y Universidad*, vol. 19, no. 1, pp. 207–228, 2015.

[26] T. Pizzuti and G. Mirabelli, "The Global Track&Trace System for food: General framework and functioning principles," *Journal of Food Engineering*, vol. 159, pp. 16–35, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jfoodeng.2015.03.001

[27] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, 2013.

[28] R. Boncea and I. Bacivarov, "A System Architecture for Monitoring the Reliability of IoT," in *Proceedings of the 15th International Conference on Quality and Dependability*, no. 143-149, 2016, pp. 143–149.

[29] M. S. Elbamby, M. Bennis, W. Saad, M. Latva-aho, and C. S. Hong, "Proactive edge computing in fog networks with latency and reliability guarantees," *EURASIP Journal on Wireless Communications and Networking*, 2018.

[30] B. Kang and H. Choo, "An experimental study of a reliable IoT gateway," *ICT Express*, vol. 4, no. 3, pp. 130–133, 2018.

[31] C. Huang, D. Liu, J. Ni, R. Lu, and X. Shen, "Reliable and privacy-preserving selective data aggregation for fog-based IoT," *IEEE International Conference on Communications*, vol. 2018-May, pp. 1–6, 2018.

[32] U. Kulau, S. Rottmann, S. Schildt, J. Van Balen, and L. Wolf, "Undervolting in real world WSN applications: A long-term study," *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 9–16, 2016.

[33] F. Carpio, A. Jukan, A. I. Martin Sanchez, N. Amla, and N. Kemper, "Beyond Production Indicators: A Novel Smart Farming Application and System for Animal Welfare," *Proceedings of the Fourth International Conference on Animal-Computer Interaction*, pp. 7:1—-7:11, 2017.

[34] J. Tummers, A. Kassahun, and B. Tekinerdogan, "Reference architecture design for farm management information systems: a multi-case study approach," *Precision Agriculture*, vol. 22, pp. 22–50, 2021.

[35] E. Duncan, A. Glaros, D. Z. Ross, and E. Nost, "New but for whom? discourses of innovation in precision agriculture," *Agriculture and Human Values*, pp. 1–19, 2021.

[36] C. Sørensen, S. Fountas, E. Nash, L. Pesonen, D. Bochtis, S. M. Pedersen, B. Basso, and S. Blackmore, "Conceptual model of a future farm management information system," *Computers and electronics in agriculture*, vol. 72, no. 1, pp. 37–47, 2010.

[37] N. Novkovic, C. Huseman, T. Zoranovic, and B. Mutavdzic, "Farm management information systems." in *HAICTA*, 2015, pp. 705–712.

[38] S. Nandurkar, V. Thool, and R. C. Thool, "Design and development of precision agriculture system using wireless sensor network," in *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*. IEEE, 2014, pp. 1–6.

[39] R. Bongiovanni and J. Lowenberg-DeBoer, "Precision agriculture and sustainability," *Precision agriculture*, vol. 5, no. 4, pp. 359–387, 2004.

[40] M. Cicioğlu and A. Çalhan, "Smart agriculture with internet of things in cornfields," *Computers & Electrical Engineering*, vol. 90, p. 106982, 2021.

[41] M. Saravanan and S. K. Perepu, "Focusing social media based analytics for plant diseases in smart agriculture," in *Proceedings of the 5th Multidisciplinary International Social Networks Conference*. ACM, 2018, p. 20.

[42] B. Archer *et al.*, "Demand forecasting and estimation." *Demand forecasting and estimation.*, pp. 77–85, 1987.

[43] F. TongKe, "Smart agriculture based on cloud computing and iot," *Journal of Convergence Information Technology*, vol. 8, no. 2, 2013.

[44] A. Patil, M. Beldar, A. Naik, and S. Deshpande, "Smart farming using arduino and data mining," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2016, pp. 1913–1917.

[45] F. Viani, F. Robol, M. Bertolli, A. Polo, A. Massa, H. Ahmadi, and R. Boualleague, "A wireless monitoring system for phytosanitary treatment in smart farming applications," in *2016 IEEE International Symposium on Antennas and Propagation (APSURSI)*. IEEE, 2016, pp. 2001–2002.

[46] Amandeep, A. Bhattacharjee, P. Das, D. Basu, S. Roy, S. Ghosh, S. Saha, S. Pain, S. Dey, and T. K. Rana, "Smart farming using IOT," *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2017*, pp. 278–280, 2017.

[47] A. Kamilaris, F. Gao, F. X. Prenafeta-Boldu, and M. I. Ali, "Agri-IoT: A semantic framework for Internet of Things-enabled smart farming applications," *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*, pp. 442–447, 2017.

[48] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (iot) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[49] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[50] G. M. Lee, J. Park, N. Kong, and N. Crespi, "The internet of things: concept and problem statement: 01," Ph.D. dissertation, Dépt. Réseaux et Service Multimédia Mobiles (Institut Mines-Télécom-Télécom ..., 2011.

[51] IEEE *et al.*, "Towards a definition of the internet of things (iot)," *Revision-1, on-line: http://iot. ieee. org/images/-files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_ Revision1_27MAY15. pdf. Accessed*, vol. 27, no. 2017, pp. 479–501, 2015.

[52] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Computer Networks*, vol. 143, pp. 221–246, 2018.

[53] X. Wang and L. Nannan, "The application of internet of thingsin agricultural means of production supply chain management," *WIT Transactions on Information and Communication Technologies*, no. 7, 2014.

[54] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, "In-

ternet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming: a comprehensive review," *Internet of Things*, p. 100187, 2020.

[55] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[56] J. A. Manrique, J. S. Rueda-Rueda, and J. M. Portocarrero, "Contrasting Internet of Things and Wireless Sensor Network from a Conceptual Overview," *Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data*, pp. 252–257, 2017.

[57] S. E. Díaz, J. C. Pérez, A. C. Mateos, M.-C. Marinescu, and B. B. Guerra, "A novel methodology for the monitoring of the agricultural production process based on wireless sensor networks," *Computers and electronics in agriculture*, vol. 76, no. 2, pp. 252–265, 2011.

[58] O. Debauche, J.-P. Trani, S. Mahmoudi, P. Manneback, J. Bindelle, S. Mahmoudi, and F. Lebeau, "Data management and internet of things: A methodological review in smart farming," *Internet of Things*, p. 100378, 2021.

[59] S. J. Moore, C. D. Nugent, S. Zhang, and I. Cleland, "Iot reliability: a review leading to 5 key research directions," *CCF Transactions on Pervasive Computing and Interaction*, pp. 1–17, 2020.

[60] J. Radatz, A. Geraci, and F. Katki, "Ieee standard glossary of software engineering terminology," *IEEE Std*, vol. 610121990, no. 121990, p. 3, 1990.

[61] S. C. o. t. I. R. Society, *IEEE Recommended Practice on Software Reliability*, 2008, no. June.

[62] L. Li, Z. Jin, G. Li, L. Zheng, and Q. Wei, "Modeling and analyzing the reliability and cost of service composition in the iot: A probabilistic approach," in *2012 IEEE 19th International Conference on Web Services*. IEEE, 2012, pp. 584–591.

[63] J. L. Romeu, "Understanding Series and Parallel Systems Reliability," *Reliability Analysis Center*, vol. 11, no. 5, p. 8, 2012.

[64] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of internet of things in smart agriculture: A survey," *Future Generation Computer Systems*, vol. 126, pp. 169–184, 2022.

[65] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE access*, vol. 8, pp. 85 714–85 728, 2020.

[66] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, Trends, and Prospects in Edge Technologies: Fog, Cloudlet, Mobile Edge, and Micro Data Centers," *Computer Networks*, 2017.

[67] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big data and internet of things: A roadmap for smart environments*. Springer, 2014, pp. 169–186.

[68] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," in *Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), 2015 2nd International Symposium on*. IEEE, 2015, pp. 1–5.

[69] H. Atlam, R. Walters, and G. Wills, "Fog Computing and the Internet of Things: A Review," *Big Data and Cognitive Computing*, vol. 2, no. 2, p. 10, 2018.

[70] M. T. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proc. of the Sixth International Conference on Advances in Future Internet*. Citeseer, 2014, pp. 48–55.

[71] G. I. Klas, "Fog computing and mobile edge cloud gain momentum open fog consortium, etsi mec and cloudlets," *Google Scholar*, 2015.

[72] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[73] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for iot," in *Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on*. IEEE, 2015, pp. 687–694.

[74] V. Avelar. (2017) Practical options for deploying small server rooms and micro data centers. [Online]. Available: http://www.datacenterresearch.com/whitepaper/practical-options-for-deploying-small-server-rooms-and-micro-9014.html

[75] B. Brown. (2017) Why micro datacenters really matter to mobile's future. [Online]. Available: http://www.networkworld.com/article/2979570/cloud-computing/microsoft-researcher-why-micro-datacenters-really-matter-to-mobiles-future.html

[76] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[77] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2017.

[78] H. R. Abdulqadir, S. R. Zeebaree, H. M. Shukur, M. M. Sadeeq, B. W. Salim, A. A. Salih, and S. F. Kak, "A study of moving from cloud computing to fog computing," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 60–70, 2021.

[79] P. Maiti, J. Shukla, B. Sahoo, and A. K. Turuk, "Qos-aware fog nodes placement," in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*. IEEE, 2018, pp. 1–6.

[80] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[81] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)*. IEEE, 2015, pp. 73–78.

[82] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for iot," *IEEE Access*, vol. 6, pp. 115–124, 2018.

[83] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," in *Internet of things*. Elsevier, 2016, pp. 61–75.

[84] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," pp. 37–42, 2015. [Online]. Available: http://doi.acm.org/10.1145/2757384.2757397{%}5Cnhttp://dl.acm.org/ft{_}gateway.cfm?id=2757397{&}type=pdf

[85] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1–8.

[86] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things : a primer," *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, 2018.

[87] X. He, Z. Ren, C. Shi, and J. Fang, "A novel load balancing strategy of software-defined cloud/fog networking in the internet of vehicles," *China Communications*, vol. 13, no. 2, pp. 140–149, 2016.

[88] K. Liang, L. Zhao, X. Chu, and H.-H. Chen, "An integrated architecture for software defined and virtualized radio access networks with fog computing," *IEEE Network*, vol. 31, no. 1, pp. 80–87, 2017.

[89] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, 2018.

[90] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: A study of infrastructure as a service (iaas)," *International Journal of engineering and information Technology*, vol. 2, no. 1, pp. 60–63, 2010.

[91] E. Keller and J. Rexford, "The" platform as a service" model for networking." *INM/WREN*, vol. 10, pp. 95–108, 2010.

[92] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24–31, 2013.

[93] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[94] Y. Xiao and M. Krunz, "Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[95] D. Bendouda, A. Rachedi, and H. Haffaf, "An hybrid and proactive architecture based on sdn for internet of things," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*. IEEE, 2017, pp. 951–956.

[96] M. Ibrar, L. Wang, G.-M. Muntean, J. Chen, N. Shah, and A. Akbar, "Ihsf: An intelligent solution for improved performance of reliable and time-sensitive flows in hybrid sdn-based fc iot systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3130–3142, 2020.

[97] A. I. Montoya-Munoz, D. M. Casas-Velasco, F. Estrada-Solano, A. Ordonez, and O. M. C. Rendon, "A yang model for a vertical sdn management plane," in *2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, 2017, pp. 1–6.

[98] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM computer communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[99] F. Popentiu-Vladicescu and G. Albeanu, "Software reliability in the fog computing," in *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*. IEEE, 2017, pp. 1–4.

[100] M. Baghrous, A. Ezzouhairi, and N. Benamar, "Smart farming system based on fog computing and lora technology," *Embedded Systems and Artificial Intelligence: Proceedings of ESAI 2019, Fez, Morocco*, vol. 1076, p. 217, 2020.

[101] A. Brogi, S. Forti, A. Ibrahim, and L. Rinaldi, "Bonsai in the Fog: An active learning lab with Fog computing," *2018 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018*, pp. 79–86, 2018.

[102] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of iot in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156 237–156 271, 2019.

[103] C. Arivazhagan and V. Natarajan, "A survey on fog computing paradigms, challenges and opportunities in iot," in *2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2020, pp. 0385–0389.

[104] M. Aazam and E.-N. Huh, "Fog computing: The cloud-iot\/ioe middleware paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, 2016.

[105] S. J. Moore, C. D. Nugent, S. Zhang, and I. Cleland, "IoT reliability: a review leading to 5 key research directions," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 3, pp. 147–163, 2020. [Online]. Available: https://doi.org/10.1007/s42486-020-00037-z

[106] N. Omar, H. Zen, N. N. A. A. A. Aldrin, W. Waluyo, and F. Hadiatna, "Accuracy and reliability of data in iot system for smart agriculture," *International Journal of Integrated Engineering*, vol. 12, no. 6, pp. 105–116, 2020.

[107] P. A. Londra, I.-E. Kotsatos, N. Theotokatos, A. T. Theocharis, and N. Dercas, "Reliability analysis of rainwater harvesting tanks for irrigation use in greenhouse agriculture," *Hydrology*, vol. 8, no. 3, p. 132, 2021.

[108] K. Wang, Y. Shao, L. Xie, J. Wu, and S. Guo, "Adaptive and Fault-tolerant Data Processing in Healthcare IoT Based on Fog Computing," *IEEE Transactions on Network Science and Engineering*, pp. 1–11, 2018.

[109] A. Akbar, M. Ibrar, M. A. Jan, A. K. Bashir, and L. Wang, "Sdn-enabled adaptive and reliable communication in iot-fog environment using machine learning and multiobjective optimization," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3057–3065, 2020.

[110] R. A. C da Silva and N. L. S da Fonseca, "On the location of fog nodes in fog-cloud infrastructures," *Sensors*, vol. 19, no. 11, p. 2445, 2019.

[111] T. Lähderanta, T. Leppänen, L. Ruha, L. Lovén, E. Harjula, M. Ylianttila, J. Riekki, and M. J. Sillanpää, "Edge computing server placement with capacitated location allocation," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 130–149, 2021.

[112] H. D. Chantre and N. L. S. da Fonseca, "The location problem for the provisioning of protected slices in nfv-based mec infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1505–1514, 2020.

[113] H. D. Chantre and N. L. da Fonseca, "Multi-Objective Optimization for Edge Device Placement and Reliable Broadcasting in 5G NFV-Based Small Cell Networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2304–2317, 2018.

[114] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, "IoT and agriculture data analysis for smart farm," *Computers and Electronics in Agriculture*, vol. 156, no. December 2018, pp. 467–474, 2019. [Online]. Available: https://doi.org/10.1016/j.compag.2018.12.011

[115] R. Pahuja, H. Verma, and M. Uddin, "A wireless sensor network for greenhouse climate control," *IEEE Pervasive Computing*, vol. 12, no. 2, pp. 49–58, 2013.

[116] A. Natale, S. Antognelli, E. Ranieri, A. Cruciani, and A. Boggia, "A novel cleaning method for yield data collected by sensors: A case study on win-

ter cereals," in *International Conference on Computational Science and Its Applications*. Springer, 2020, pp. 684–691.

[117] 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, 2017.

[118] M. Taneja, N. Jalodia, J. Byabazaire, A. Davy, and C. Olariu, "SmartHerd management: A microservices-based fog computing–assisted IoT platform towards data-driven smart dairy farming," *Software - Practice and Experience*, vol. 49, no. 7, pp. 1055–1078, 2019.

[119] M. Taneja, J. Byabazaire, N. Jalodia, A. Davy, C. Olariu, and P. Malone, "Machine learning based fog computing assisted data-driven approach for early lameness detection in dairy cattle," *Computers and Electronics in Agriculture*, vol. 171, no. February, p. 105286, 2020. [Online]. Available: https://doi.org/10.1016/j.compag.2020.105286

[120] T. C. Hsu, H. Yang, Y. C. Chung, and C. H. Hsu, "A Creative IoT agriculture platform for cloud fog computing," *Sustainable Computing: Informatics and Systems*, 2018. [Online]. Available: https://doi.org/10.1016/j.suscom.2018.10.006

[121] M. Taneja, N. Jalodia, P. Malone, J. Byabazaire, A. Davy, and C. Olariu, "Connected cows: Utilizing fog and cloud analytics toward data-driven decisions for smart dairy farming," *IEEE Internet of Things Magazine*, vol. 2, no. 4, pp. 32–37, 2019.

[122] Çelik, Mete and Dadaşer-Çelik, Filiz and Dokuz, Ahmet Şakir, "Anomaly detection in temperature data using DBSCAN algorithm," *INISTA 2011 - 2011 International Symposium on INnovations in Intelligent SysTems and Applications*, no. June, pp. 91–95, 2011.

[123] Z. Ding and M. Fei, *An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window*. IFAC, 2013, vol. 3. [Online]. Available: http://dx.doi.org/10.3182/20130902-3-CN-3020.00044

[124] A. Fawzy, H. M. Mokhtar, and O. Hegazy, "Outliers detection and classification in wireless sensor networks," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 157–164, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.eij.2013.06.001

[125] N. Wahir, M. Nor, M. Rusiman, and K. Gopal, "Treatment of outliers via interpolation method with neural network forecast performances," in *Journal of Physics: Conference Series*, vol. 995. IOP Publishing, 2018, pp. 1–7.

[126] L. F. Maldaner, J. P. Molin, and M. Spekken, "Methodology to filter out outliers in high spatial density data to improve maps reliability," *Scientia Agricola*, vol. 79, 2021.

[127] Q. Minh, T. Phan, A. Takahashi, T. Thanh, S. Duy, M. Thanh, and C. Hong, "A cost-effective smart farming system with knowledge base," *ACM International Conference Proceeding Series*, vol. 2017-Decem, no. December 2017, pp. 309–316, 2017.

[128] L. D.D.Chaudhary, S.P.Nayse, "Application of Wireless Sensor Networks for Greenhouse Parameter Control in Precision Agriculture," *International Journal of Wireless & Mobile Networks*, vol. 3, no. 1, p. 140, 2011.

[129] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things," *Practical Research - Planning & Design*, vol. 3, no. 1, p. 67, 2010. [Online]. Available: https://www.researchgate.net/publication/303192188{%}0Ahttp://icas-pub.org/ojs/index.php/ticc/article/view/47

[130] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021. [Online]. Available: https://www.gurobi.com

[131] Gurobi Optimization, LLC; Python Software Foundation, "gurobipy 9.1.2," 2021. [Online]. Available: https://pypi.org/project/gurobipy/

[132] Gurobi Optimization, LLC, "Gurobi Python API Overview," 2021. [Online]. Available: https://www.gurobi.com/documentation/9.1/refman/py_python_api_overview.html#sec:Python

[133] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[134] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta, "Review of IoT applications in agro-industrial and environmental fields," *Computers and Electronics in Agriculture*, vol. 142, no. September, pp. 283–297, 2017.

[135] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-Khah, and P. Siano, "Iot-based smart cities: a survey," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*. IEEE, 2016, pp. 1–6.

[136] J. Sherly and D. Somasundareswari, "Internet of things based smart transportation systems," *International Research Journal of Engineering and Technology*, vol. 2, no. 7, pp. 1207–1210, 2015.

[137] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of internet of things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017.

[138] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proceedings of the ASE BigData & SocialInformatics 2015*, 2015, no. October, pp. 1–6.

[139] H. Abukhalaf, J. Wang, and S. Zhang, "Outlier Detection Techniques for Localization in Wireless Sensor Networks: A Survey," *International Journal of Future Generation Communication and Networking*, vol. 8, no. 6, pp. 99–114, 2015.

[140] E. F. Castillo, W. F. Gonzales, I. D. López, A. Figueroa, D. C. Corrales, M. G. Hoyos, and J. C. Corrales, "Water quality warnings based on cluster analysis in colombian river basins," *Sistemas & Telemática*, vol. 13, no. 33, pp. 9–26, 2015.

[141] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[142] Y. Liu, H. Wang, H. Zhang, and K. Liber, "A comprehensive support vector machine-based classification model for soil quality assessment," *Soil and Tillage Research*, vol. 155, pp. 19–26, 2016.

[143] H. P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[144] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems*, vol. 42, no. 3, 2017.

[145] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, 1996, pp. 226–231.

[146] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[147] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *2007 IEEE International Conference on Communications*. IEEE, 2007, pp. 3864–3869.

[148] A. Pradhan, "SUPPORT VECTOR MACHINE-A Survey," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 8, pp. 82–85, 2012.

[149] L. László, "Cubic spline interpolation with quasiminimal B-spline coefficients," *Acta Mathematica Hungarica*, vol. 107, no. 1-2, pp. 77–87, 2005.

[150] M. Hazewinkel, *Encyclopaedia of mathematics: volume 6: subject index—author index*. Springer Science & Business Media, 2013.

[151] O. Rukundo, "Nearest Neighbor Value Interpolation," *arXiv preprint arXiv:1211.1768*, vol. 3, no. 4, pp. 1–6, 2012.

[152] P. Klapetek, "Basic Data Processing," *Quantitative Data Processing in Scanning Probe Microscopy*, pp. 55–80, 2013.

[153] "Libelium waspmote plug and sense!" http://www.libelium.com/products/plug-sense/technical-overview/, accessed: 2020-07-28.

[154] "Datalog x pro," https://www.omicroning.co/es/datalog-x-pro.html, accessed: 2020-07-28.

[155] R. Dawson, "How significant is a boxplot outlier?" *Journal of Statistics Education*, vol. 19, no. 2, 2011.

[156] "Reliable iot-agro: Github repository," https://github.com/aimontoya07/Reliable_IoT-Agro/tree/master, accessed: 2020-10-17.

[157] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[158] C. Ozgur, T. Colliau, G. Rogers, Z. Hughes, B. Myer-Tyson *et al.*, "Matlab vs. python vs. r," *Journal of Data Science*, vol. 15, no. 3, pp. 355–372, 2017.

[159] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Springer-Verlag London, UK, 2000, pp. 29–39.

[160] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, 2018.

[161] ——, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, pp. 1–99, 2018.

[162] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, no. February, pp. 70–90, 2018.

[163] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[164] U. E. C. of Engineering and A. Science, "Problem solving i," http://www.eas.uccs.edu/~mwickert/ece1010/lecture_notes/1010n6a.PDF,.

A Fog Computing-based mechanism to provide reliability in IoT: A case study in a coffee Smart Farming environment

ANNEXES

Master Degree work

Ana Isabel Montoya Muñoz

Advisor: PhD. Oscar Mauricio Caicedo Rendón
Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2021

Appendix A

Gurobipy-based implementation

Appendix A presents the content of the python file *fogopt.py*.

```
#!/usr/bin/env python3.7
'''
This file contains a optimization model for protecting Smart Farms by
Fog layer infrastructure reliable deployment (protected FNs by redundancy)
for providing services to farmers minimizing the costs of implementation.

Created by Ana Isabel Montoya-Munoz
Universidad del Cauca
'''
from gurobipy import *
import numpy as np
import random
from collections import defaultdict
import pandas as pd
import csv

#--- PARAMETERS (INPUT) ---

# Type of demand
Demand = 'High'
```

Appendix B

Publications

The Appendix B presents the papers developed during the elaboration of the master thesis published and submitted.

- **Ana Isabel Montoya Muñoz**, Oscar Mauricio Caicedo Rendón. **An approach based on Fog Computing for providing reliability in IoT Data Collection: A Case Study in a Colombian Coffee Smart Farm** Applied Sciences 2020.
  - Status: Published.
  - Special Issue: Computing and Artificial Intelligence.
  - Classification: A1 MinCiencias - Q2 (JCR).

- Jhonn Pablo Rodríguez, **Ana Isabel Montoya Muñoz**, Carlos Rodriguez Pabón, Javier Hoyos, and Juan Carlos Corrales. **IoT-Agro: A smart farming system to Colombian coffee farms** Computers and Electronics in Agriculture 2021.
  - Status: Published.
  - Classification: A1 MinCiencias - Q1 (JCR).

- **Ana Isabel Montoya Muñoz**, Rodrigo A. C. da Silva, Oscar Mauricio Caicedo Rendón, and Nelson L. S. da Fonseca. **Provisioning Protection to Smart Farming** Computers and Electronics in Agriculture 2021.
  - Status: Submitted.
  - Classification: A1 MinCiencias - Q1 (JCR).

- **Ana Isabel Montoya Muñoz**, Daniela Casas Velasco, Felipe Estrada Solano, Oscar Mauricio Caicedo Rendón, and Nelson L. Saldanha da Fonseca. **An approach based on Yet Another Next Generation for software-defined networking management** International Journal of Communication Systems 2021.
  - Status: Published.
  - Classification: A1 MinCiencias - Q3 (JCR).