



HAL
open science

Embedded-AI-enabled semantic IoT platform for agroecology

Nicolas Chollet

► **To cite this version:**

Nicolas Chollet. Embedded-AI-enabled semantic IoT platform for agroecology. Embedded Systems. Université Paris-Saclay, 2023. English. NNT : 2023UPASG078 . tel-04474508

HAL Id: tel-04474508

<https://theses.hal.science/tel-04474508>

Submitted on 23 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Embedded-AI-enabled semantic IoT platform for agroecology

*Plateforme IoT sémantique compatible avec l'IA
embarquée pour l'agroécologie*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580, Sciences et technologies de l'information et de la
communication (STIC)

Spécialité de doctorat: Informatique

Graduate School : Informatique et Sciences du Numérique

Référent : Université de Versailles Saint-Quentin-en-Yvelines

Thèse préparée dans les unités de recherche du **LISV** (Université Paris-Saclay, UVSQ) et du
LyRIDS (ECE), sous la direction de **Amar RAMDANE-CHERIF**, Professeur des universités et
le co-encadrement de **Naila BOUCHEMAL**, Enseignante-chercheuse

Thèse soutenue à Paris, le 30 Novembre 2023, par

Nicolas CHOLLET

Composition du jury

Membres du jury avec voix délibérative

Toufik AHMED Professeur des universités, Laboratoire Bordelais de Recherche en Informatique (LaBRI)	Président & Rapporteur
Ousmane THIARE Professeur des universités, Habilité à Diriger des Recherches (HDR), Université Gaston Berger de Saint- Louis (UGB)	Rapporteur & Examineur
Hakima CHAOUCHI Professeure, Telecom Sud Paris	Examinatrice
Sondes KHEMIRI-KALLEL Maître de conférence, Laboratoire DAVID - UVSQ	Examinatrice
Assia SOUKANE Maître de conférence, LyRIDS - ECE Paris	Examinatrice
Mikal ZIANE HDR, Maître de conférence, Laboratoire d'informatique de Sorbonne Université - LIP6	Examineur

Titre: Plateforme IoT sémantique compatible avec l'IA embarquée pour l'agroécologie.

Mots clés: Internet des Objets (IoT), Agriculture durable, Intelligence Artificielle (IA) embarquée, Réseaux de capteurs sans fils, Low Power Wide Area Networks (LPWAN), Ingénierie des connaissances.

Résumé: L'agriculture moderne nécessite une profonde transformation pour répondre aux défis du développement durable tout en nourrissant qualitativement et quantitativement la population mondiale croissante. Dans cette optique, les agriculteurs adoptent le "Smart Farming" ou agriculture intelligente. C'est une méthode agricole qui utilise la technologie pour améliorer l'efficacité, la productivité et la durabilité de la production agricole. Elle englobe l'usage de capteurs, l'internet des objets (IoT), l'Intelligence Artificielle (IA), l'analyse de données, la robotique et divers autres outils numériques optimisant des aspects tels que la gestion des sols, l'irrigation, la lutte antiparasitaire ou encore la gestion de l'élevage. L'objectif est d'augmenter la production tout en réduisant la consommation de ressources, minimisant les déchets et améliorant la qualité des produits. Toutefois, malgré ses avantages et son déploiement réussi dans divers projets, l'agriculture intelligente rencontre des limites

notamment dans le cadre de l'IoT. Premièrement, les plateformes doivent être capables de percevoir des données dans l'environnement, de les interpréter et de prendre des décisions pour aider à la gestion des fermes. Le volume, la variété et la vélocité de ces données, conjuguées à la grande diversité d'objets ainsi qu'à l'avènement de l'IA embarquée dans les capteurs, rendent difficile les communications sur les réseaux agricoles sans fils. Deuxièmement, les recherches tendent à se focaliser sur des projets répondant aux problématiques de l'agriculture conventionnelle non durable et les projets concernant les petites exploitations axées sur l'agroécologie sont rares. Dans ce contexte, cette thèse explore la création d'une plateforme IoT composée d'un réseau de capteurs intelligents sémantiques, visant à guider les agriculteurs dans la transition et la gestion de leur ferme en agriculture durable tout en minimisant l'intervention humaine.

Title: Embedded-AI-enabled semantic IoT platform for Agroecology

Keywords: Internet of Things (IoT), Sustainable Agriculture, Embedded Artificial Intelligence (AI), Wireless Sensors Networks, Low Power Wide Area Networks (LPWAN), Knowledge Engineering.

Abstract: Modern agriculture requires a profound transformation to address the challenges of sustainable development while qualitatively and quantitatively feeding the growing global population. In this light, farmers are adopting "Smart Farming" also called precision agriculture. It is an agricultural method that leverages technology to enhance the efficiency, productivity, and sustainability of agricultural production. This approach encompasses the use of sensors, the Internet of Things (IoT), Artificial Intelligence (AI), data analysis, robotics, and various other digital tools optimizing aspects such as soil management, irrigation, pest control, and livestock management. The goal is to increase production while reducing resource consumption, minimizing waste, and improving product quality. However, despite its benefits and successful deployment in vari-

ous projects, smart agriculture encounters limitations, especially within the context of IoT. Firstly, platforms must be capable of perceiving data in the environment, interpreting it, and making decisions to assist in farm management. The volume, variety, and velocity of those data, combined with a wide diversity of objects and the advent of AI embedded in sensors, make communication challenging on wireless agricultural networks. Secondly, research tends to focus on projects addressing the issues of non-sustainable conventional agriculture, and projects related to small-scale farms focused on agroecology are rare. In this context, this thesis explores the creation of an IoT platform comprised of a network of semantic smart sensors, aiming to guide farmers in transitioning and managing their farm sustainably while minimizing human intervention.

Information sur les organismes d'accueil

Organisme 1: LISV - Laboratoire d'Ingénierie des Systèmes de Versailles

Adresse: 10 Av. de l'Europe, 78140 Vélizy-Villacoublay

URL: www.lisv.uvsq.fr

Présentation:

Le LISV développe ses activités de recherche multi-disciplinaires, tant théoriques qu'expérimentales, principalement autour de la conception mécanique, l'automatique, la compréhension de phénomènes tels que la locomotion, la manipulation, l'ontologie et l'automatique pour l'équipe Robotique Interactive, et autour de la caractérisation, tant amont qu'aval, du comportement des systèmes ce qui part de la modélisation des comportements des composants afin d'appréhender le rôle des paramètres environnementaux, puis l'instrumentation et la métrologie du composant ou du système afin de valider ses performances. Les domaines d'applications principaux sont la robotique, la cobotique, la mécanique, l'assistance aux personnes et à la mobilité, les systèmes embarqués, le véhicule autonome, les capteurs et la transmission d'information. LE LISV a également de nombreuses activités de valorisation (prototypes, brevets, logiciels, 3 startup ayant émergé dans les dernières années).



Organisme 2: ECE - École Centrale d'Électronique

Adresse: 10 Rue Sextius Michel, 75015 Paris XV

URL: www.ece.fr

Présentation:

L'ECE est une école d'ingénierie française spécialisée dans le numérique, située à Paris. Elle a été fondée en 1919 sous le nom d'École centrale de la TSF dans le but de former les tout premiers opérateurs radio, puis s'est spécialisée dans les nouvelles technologies de l'information et de la communication pendant la deuxième moitié du XX^e siècle.

Le centre de recherche de l'ECE existe depuis 2004. Il compte aujourd'hui une vingtaine de permanents et une dizaine de doctorants. Il déploie une stratégie innovante qui enrichit la pédagogie et apporte aux élèves un éclairage sur les enjeux scientifiques, sociétaux et environnementaux. Les thèmes de recherche du centre couvrent un large spectre de domaines scientifiques, incluant à la fois des aspects fondamentaux et des aspects appliqués. Les travaux se développent en trois axes à la confluence des Systèmes Intelligents Communicants, des Méthodes Mathématiques pour l'Ingénierie Scientifique et Financière, et des Nanosciences et Nanotechnologies.

Depuis 2018, les étudiants de l'école ont mis en place un potager urbain sur le toit de leur campus afin de sensibiliser aux problématiques agricoles actuelles et également d'expérimenter leurs projets. Ce jardin est mis à contribution dans cette thèse, sur laquelle a été déployé un réseau de capteurs.



Acknowledgements

I would like to take a moment to express my deepest gratitude to the teachers, friends, and family who have supported and guided me throughout my PhD. Your unwavering care, patience, and encouragement has been a constant source of inspiration and motivation for me.

To my advisors, Amar and Naila, thank you for sharing your knowledge and wisdom with me. Your belief in my abilities despite some troublesome periods allowed me to hold on and pursue this quest for knowledge which is the PhD adventure. I also want to thank all the members of the lab at ECE especially its director, Assia Soukane, who was also full of support and understanding throughout those years.

To my friends, thank you for being a constant source of laughter, comfort, and support. Your companionship has helped me through the toughest of times and has made the best moments even more memorable. Your loyalty and kindness have touched my heart and I am forever grateful for your friendship. I have a particular thought for my co-PhD students at ECE Paris especially the other doctoral students, Quentin, Abderaouf, Cécile, Maxime, Alexandre, Rémi, Melchior and Cécile with whom I had the chance to fight countless battles.

To my family, thank you for being my rock and my safe haven. Your love and support have sustained me through the ups and downs of life. Your guidance and encouragement have given me the strength to overcome challenges and pursue my passions despite the mountains of doubts that were in the way. I am blessed to have such a wonderful family and I love you all dearly.

To Siobhan, with all my love, thank you. You were a late addition to the team but I could not have won it without you as your presence gave me the strength and motivation to finish the last miles.

In conclusion, to all of you, I offer my sincerest gratitude for the impact you have had on my life. Your presence has made a difference and I am grateful for all that you have done for me. Thank you for being a part of my journey and for helping me become the person I am today. Throughout my life, I will never forget what you did for me and you will always accompany me in my future adventures.

Table of Contents

1	General Introduction	13
1.1	Context and motivation	13
1.2	Problem statement	28
1.3	Methodology and Contribution	30
1.4	Report Outline	32
2	State of the art	34
2.1	Introduction	34
2.2	IoT platform	35
2.2.1	Definition	35
2.2.2	Agricultural Application	41
2.2.3	Limitations	50
2.2.4	Amelioration axes	53
2.3	Energy efficient Wireless Sensor Network	54
2.3.1	Definition	54
2.3.2	WSN communication protocols	59
2.3.3	Low Power Wide Area Network	71
2.3.4	Agricultural Application	74
2.3.5	Limitations	77
2.3.6	Amelioration axes	77
2.4	Embedded intelligence	78
2.4.1	Definition	78
2.4.2	Application	87
2.4.3	Limitations	88
2.4.4	Amelioration axes	90
2.5	Knowledge engineering	90
2.5.1	Definition	91
2.5.2	Internet of things application	94
2.5.3	Agricultural Application	95
2.5.4	IoT and agricultural Application	96
2.5.5	Limitations	97
2.5.6	Amelioration axes	98
2.6	Conclusion and proposed contribution	98
3	An energy-efficient wireless sensor network for Embedded AI devices	100
3.1	Introduction	100
3.2	Architecture proposal	101
3.2.1	Scenario	101

3.2.2	Environment development	104
3.2.3	Phases	122
3.3	Experimentation and results	127
3.3.1	Embedded AI model	127
3.3.2	Data offloading	136
3.3.3	Firmware update over the air	139
3.4	Conclusion	141
4	A knowledge-based approach for IoT platform management in agroecological farm	144
4.1	Introduction	144
4.2	Definition	145
4.2.1	Ontology	145
4.2.2	Ontology development with OWL and Protégé	147
4.2.3	Reasoning	148
4.3	Proposed model	149
4.3.1	Crops: Plant and Agroecology knowledge base	150
4.3.2	WSN: Sensors and actuators knowledge base	153
4.3.3	Farm	155
4.3.4	Rule Based System	157
4.4	Experimentation	160
4.4.1	Crops layout	160
4.4.2	Sensors management	160
4.4.3	Irrigation routine	161
4.4.4	Pest and disease detection and management	161
4.4.5	Optimal seeding and harvest detection	161
4.5	Conclusion	162
5	General Conclusion	163
5.1	General conclusion	163
5.2	Future works	164
5.3	Peer-reviewed publication	166
5.4	Additional training	168
5.5	Future opportunities	173
6	Annexe : Résumé de thèse en Français	174

List of Figures

1.1	Projected change in future global crop yields between 2010 and 2050 [19]	17
1.2	Dietary changes in developing countries, 1964-66 to 2030 [23]	18
1.3	Global land use for food production [26]	19
1.4	Development road-map of industrial revolutions and agricultural revolutions [34]	20
1.5	Smart Farming domains	21
1.6	IoT platform example	22
1.7	Edge computing interest (Google trends 2013-2020) [38]	22
1.8	Cloud computing (top) vs Edge computing (bottom)	23
1.9	Share of farms worldwide, by land size class. [59]	26
1.10	Three sisters method representation	27
1.11	Overall view of the IoT platform	29
1.12	Methodology and contribution overall view	30
1.13	IoT platform specific view	32
2.1	The cyber-physical management cycle of Smart Farming enhanced by cloud-based event and data management [69]	34
2.2	IoT platform model	36
2.3	IoT platform for livestock stress management	37
2.4	3 layers architecture	39
2.5	The internet of Things Market 2019-2030 (number of devices in billions and USD value in millions)	42
2.6	Typical Wireless sensor networks architecture	54
2.7	WSNs topologies [204]	56
2.8	OSI and TCP/IP	60
2.9	Range vs speed and energy consumption in different types of Network	70
2.10	Cloud, Fog, Edge data processing architecture	79
2.11	Neural Network architecture	84
2.12	Neural Network training process [281]	85
2.13	Neurons pruning process	86
2.14	Knowledge Pyramid	91
2.15	Expert System Basic Architecture	92
3.1	System in normal behavior mode	102
3.2	System in degraded condition mode : Data offload and FUOTA process	103
3.3	Chirp Spread Spectrum modulation of a LoRaWAN frame	105
3.4	A typical LoRaWAN network architecture	106
3.5	Comparison of LoRa Spreading Factors: SF7 to SF12	108
3.6	Class A receive windows	110
3.7	Class B receive windows	110

3.8	Class C receive windows	110
3.9	Theoretical maximum number of large media files transferable per day based on data rate and media size	112
3.10	FUOTA architecture	117
3.11	Arduino Portenta H7 and its LoRa Vision Shield	118
3.12	STM32 Nucleo pack LoRa	119
3.13	ESP32 Cam and the attached Hope RFM95W module for LoRa connectivity	119
3.14	Laird Sentrius RG168 Gateway	120
3.15	Overall process flowchart	122
3.16	Image data acquired through the Arduino Portenta are on the right, and through the ESP32 cam are on the left	123
3.17	TinyML model creation process with edge impulse	124
3.18	FUOTA Session	128
3.19	Test Bench Featuring the ESP32-Cam on the left, the Arduino Portenta in the middle, and the STM32 on the right	129
3.20	Battery Lifetime (in Days) for Various Data Rates, Based on Experimental Results and Theoretical Values	132
3.21	Simulation Outcomes for a One-Year Operational Period	133
3.22	Evolution of Battery level over time	135
3.23	Battery Lifetime of the system in days	136
3.24	Energy consumption of the system for transferring various numbers of images using the ESP32.	138
3.25	Energy consumption of the system for transferring various numbers of images using the Arduino Portenta.	138
3.26	Evolution of Battery level over time	140
3.27	Battery lifetime of the system in days	141
3.28	Evolution of Battery level over time for various size of firmware	141
3.29	Battery lifetime of the system in days for various size of firmware	142
4.1	Ontology and instanced data comparison	146
4.2	General Knowledge-Base model	150
4.3	Ontology Fairness Evaluator Results	150
4.4	Ontology Structure in Protégé	151
4.5	Relationship between different individuals of the Plant class and the inferred association based on the provided and required properties	153
4.6	Ontological representation of the Crops class	154
4.7	Ontological representation of the WSN class	155
4.8	Ontological representation of the Farm class	157
5.1	ICFUN presentation	166
5.2	ITNAC presentation	167
5.3	6gnet presentation	167
5.4	Eceborg group photo when I was president	171

5.5	Vegetable garden of the school	172
5.6	Class teaching	172

List of Tables

1.1	Types of environmental degradation, causes and impacts on agricultural processes . . .	17
2.1	Usual technologies used in the IoT domain	40
2.2	Main network protocols according to OSI/ISO Model used in the IoT field	41
2.3	Main LPWANs technical specificities	75
3.1	Bit rate for each data rate (DR0 - DR6) configured with the spreading factor and the bandwidth.	107
3.2	Packet sizes (N) in bytes for each Data Rate of European regulation	107
3.3	Difference in bit rate according to bandwidth	109
3.4	Image transfer via LoRa literature survey	114
3.5	Hardware and Software Components for System Evaluation	121
3.6	Hardware Characteristics	129
3.7	Hardware deployment results	130
3.8	Transmission Time in Milliseconds of Inference Result for Various Data Rates and maximal number of inference a day according to Duty Cycle	131
3.9	Power Consumption for Each Device Across Various States	131
3.10	Simulation Parameters	134
3.11	Total Energy Requirement in mA for One Year of Device Operation	135
3.12	Average Image Transmission Time (s) and Maximum Daily Transfers at Various Data Rates, According to Duty Cycle.	137
3.13	Number of Image Offload Operations Possible with a 2000 mA Battery, According to Data Rate and Quantity of Images Sent in a Single Operation.	139
3.14	FUOTA process duration in seconds (s)	139
4.1	SWRL rules	159
5.1	Mandatory additional training of the Doctoral School of Paris-Saclay	168
5.2	Optional activities and trainings	170

List of Abbreviations

- **AI** : Artificial Intelligence
- **FUOTA**: Firmware Update Over The Air
- **ICT** : Information and Communication Technology
- **IoT** : Internet of Things
- **IIoT** : Industrial Internet of Things
- **LPWAN** : Low Power Wide Area Network
- **ML** : Machine Learning
- **PA** : Precision Agriculture
- **QoS** : Quality of Service
- **SF** : Smart Farming
- **SSF** : Small-Scale Farm (and Farmer)
- **TinyML** : Tiny Machine Learning
- **WSN** : Wireless Sensor Network

1 - General Introduction

The introductory chapter of this report provides a comprehensive overview in four distinct sections, namely context and motivation, problem statement, methodology and contribution, and report outline. The context and motivation section highlights the reasons behind the chosen topic, while the problem statement section elaborates on the specific problem identified within the broader context. The methodology and contribution section discusses the process of research involved and explains the relevance of the study. Lastly, the report outline presents all the chapters in detail.

1.1 . Context and motivation

Agriculture is the foundation of human civilization as its invention mark the end of the hunter and gathering period and the creation of cities as our ancestors went from a nomadic life style to a sedentary one. Throughout our history, agriculture was a labour-intensive system with low productivity and a high sensitivity to climatic event. It was able to feed the population but required a vast number of small farms and at least a third of the population to be active in the primary agricultural production process. During the industrialisation period at the end of XIXth and the beginning of the XXth century, agriculture was reinvented. This new modern agriculture, also called agriculture 2.0 or the green Revolution, is characterized by the massive usage of heavy machinery, chemicals fertilizers and chemical protection (insecticide and pesticide). Nowadays, it is the main agricultural paradigm in the world as it allows farmer to grow important quantity of food for minimal human effort and strong adaptability to climate-induced constraints. Therefore, modern agriculture has been a major contributor to the world's food supply and has played a crucial role in supporting the growing global population [1]. However, the practice of modern agriculture has also given rise to a host of challenges that pose serious risks to the environment, health, and economic sustainability of the industry threatening humankind food security[2]. Food security is the state where all individuals in a population have physical, social, and economic access to sufficient, safe, and nutritious food to meet their dietary needs and food preferences for an active and healthy life. It involves not only the availability of food but also the accessibility, affordability, and quality of food [3].

One of the primary issues of modern agriculture is its impact on the environment . Large-scale farming practices have led to soil degradation, water pollution, and deforestation [4]. Modern agriculture is heavily dependent

on synthetic fertilizers, pesticides, and herbicides, which have a negative impact on the environment. The widespread use of these chemicals has led to soil and water contamination, and has also resulted in the development of pesticide-resistant insects and weeds [5]. Moreover, chemicals usage in the food chain might be the causes on numerous human diseases and hormonal disorders [6]. In the meantime, the extended cultivated area surfaces require the usage of heavy machinery that not only uses polluting fossil fuel powered motors but also tends to increase soil compaction due to their weight [7]. Another environmental issue of modern agriculture is the mono-cropping or monoculture. This method, where farmer use only one variety of crops per field to facilitate the harvest and the cultivation process, induce a decline in soil fertility as the same plants will need the same nutrients season after season, thus requiring more fertilizer usage and depleting the soil fertility. Moreover monoculture also increase the risk of diseases and pest development therefore inducing more pesticide usage [8]. A comprehensive list of environmental consequences caused by traditional modern Agricultural practices is proposed in Table 1.1.

Cause

Type	Cause	Degradation process	Impact
Physical	Deforestation [9]	Breakdown of natural soil structure, aggregation and porosity	Reduction in infiltration capacity. Changes in soil water-retention characteristics. Diminish earth capacity to absorb carbon based gases and produce dioxygen.
	Excessive human, livestock and machine traffic [7]	Soil compaction, breakdown of natural soil structure	Decreases infiltration capacity. Changes in soil water-retention characteristics Block aeration for soil biota. Limit roots spreading capacities.
	Excessive tillage [10]	Compaction of surface and subsoil, reduction in proportion and strength/stability of aggregates	Accelerated erosion by water and wind. Increase in bulk density leading to reduction in porosity Water logging and anaerobiosis

	Excessive water usage [11]	Soil Erosion and diminution of ground water level	Loss of soil structure. Dilution of organic matter and diminution of soil fertility. Depletion of natural ground water stock impacts the natural water cycle inducing drought and flood and reducing water availability for future crops.
	Intensive urbanisation [12]	Destruction of natural soil	Loss of soil fertility. Usage of other natural zone for waste disposal. Loss of soil infiltration capacities due to concrete coating.
Chemical	Little to no use of fertilizers [13]	Nutrient depletion	Decreased levels of macronutrients on exchange sites, soil organic matter, and in soil solution
	Excessive usage of fertilizers [13]	Acidification, eutrophication and population exposure	Leaching and runoff of nutrients to water sources polluting them in the long term.
	Usage of pesticide [5]	Soil and plants contamination	Exposure to chemical potentially harmful to human, either directly (during spraying of the field) or indirectly (inclusion of chemicals in the food produces). Creating sterner diseases and pest strains with stronger resistance to the pesticides

	Usage of insecticide [5]	Depletion of insects population, exposure to population	Disturbance of the natural food chain inducing disappearance of some species (e.g birds for who insects are the main food source) and over representation of other (e.g diminution of spider induce a an increase number of mosquitoes); Exposure to chemical potentially harmful to human , either directly (during spraying of the field) or indirectly (inclusion of chemicals in the food produces)
	Usage of poor quality water and inadequate drainage[11]	Salinization, alkalinization	Loss in soil fertility.
	Industrial and urban wastes [14]	Toxification, contamination with heavy metals, pollution	Excessive build up of some elements (e.g., Al, Mn, Fe) and heavy metals (e.g., lead and mercury); increase in soilborne pathogens
Biological	Removal of residues (leftover from crops after harvest) [15]	Nutrient depletion and soil structure degradation	Reduction of soil fertility, soil aggregation, and related properties
	Little to no use of organic input [13]	Decline in diversity and abundance of soil biota	Decrease of soil fertility level

	Introduction of not adapted species in the biota [16]	Unfair natural competition leading to some species being surepresented and other underrepresented compare to the natural biota of the ecosystem	Crops beeing more sensitive to some pest and disease. Crops become more used as food source by some species.
--	---	---	--

Table 1.1: Types of environmental degradation,causes and impacts on agricultural processes

Besides its harmful method to the environment, modern agriculture is also not suited to face climate change. There is a subsequent number of research on the effect of Climate change that shows that the earth is getting warmer since the middle of the XIX century due to human activity and its greenhouse gasses production [17]. This rise in temperature, even if contained to a few degrees °C, will disrupt the rainfall pattern as well as increase the frequency and hardship of extreme weather condition across the world such has flooding , heatwave and fire or monsoon . This will induce an important and unequal shift in climate pattern across the world. As we can see in Figure 1.1 some region will be subject to a drastic reduction in their food production capability reducing the food security of their population especially in developing countries where smallholder farmers are more affected by the weather conditions [18].

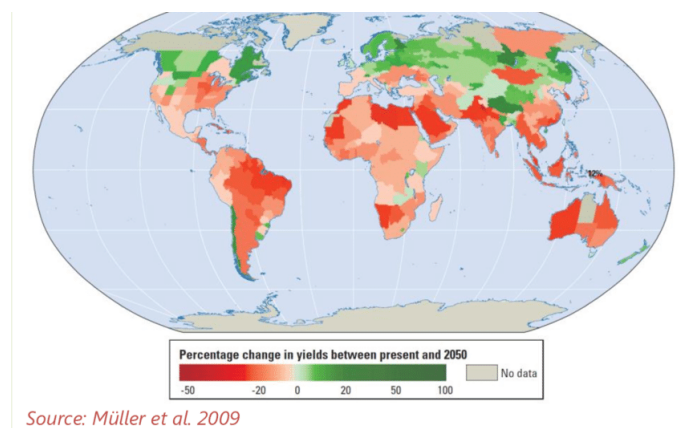


Figure 1.1: Projected change in future global crop yields between 2010 and 2050 [19]

Another issue related to modern agriculture is the impact of population growth and its impact on food consumption. The world's population is projected to reach 9.7 billion by 2050 [20], and this will put an enormous strain on the global food supply[21]. As the population continues to grow, the demand for food will increase, and farmers will need to produce more food to meet the needs of the population. Moreover, the standard of living of humankind is, for the better, rising, inducing more food consumption and an increased demand for higher grade food product (e.g. animal products) from developing countries population as shown in figure 1.2 [22].

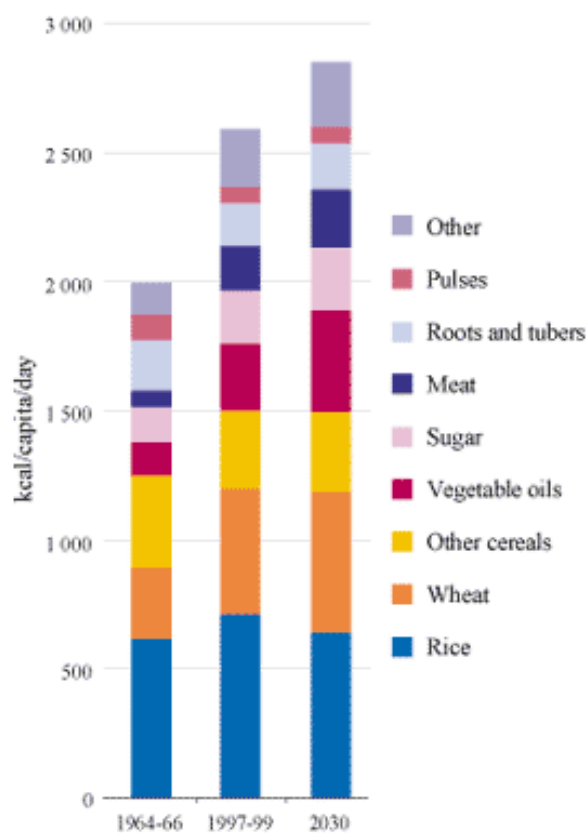


Figure 1.2: Dietary changes in developing countries, 1964-66 to 2030 [23]

The Food and Agriculture Organization (FAO) of the United States therefore estimate that by 2050, food production must increase by around 60% if we want to insure global food security to provide for everyone's needs in terms of quantity and quality and availability [24]. With the current agricultural methods this would mean increasing the agricultural land by more than 593 million hectares, that is to say twice the size of India [21]. Regrettably, as illustrated in Figure 1.3, Earth can no longer accommodate an expansion in farming areas. Over 50% of habitable land is currently allocated to agriculture, and any further expansion would jeopardize our planet's ecosystems. It's

also imperative for readers to recognize from this figure that while technology and Smart Farming (SF) offer significant benefits (as it will be later showcased in the thesis), achieving sustainable agriculture also requires a reduction in livestock-derived products consumption. Indeed, livestock farming occupies a staggering 80% of all farmland while only accounting for approximately less than 1/3 of our global calorie and protein supply that could be easily replaced by plant-based produces [25].

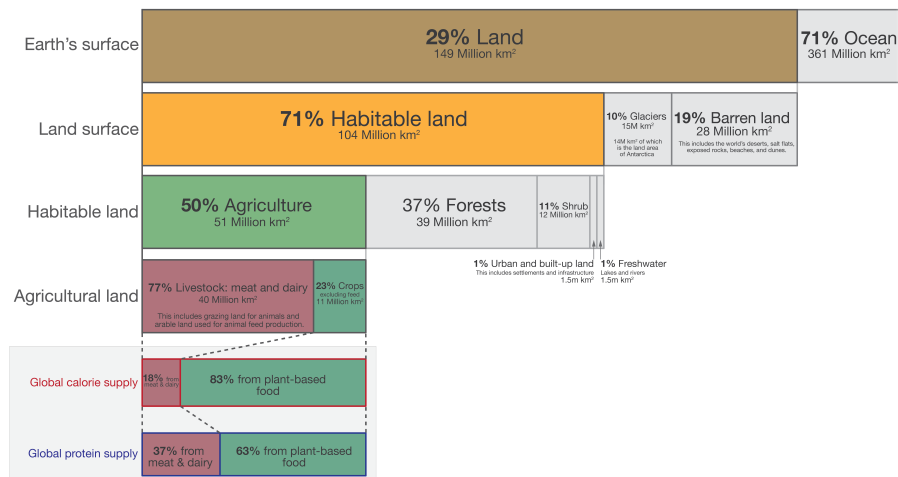


Figure 1.3: Global land use for food production [26]

In addition of the environmental and population issues, agriculture is facing also social problems in the form of a shortage of skilled labor. The average age of farmers in many countries is over 60, and there are fewer young people entering the industry as wages are low and work conditions are harsh [27]. This shortage of skilled labor is making it difficult for farmers to make a change and adopt new sustainable agricultural practices. Moreover aging farmers are more reluctant to adopt the usage of new technologies, which are essential for increasing efficiency and productivity [28]. In a more global way, older farmers usually own their farmland and sell them when going into retirement. Currently it is mainly big company with financial interest that tends to buy those land to create huge farm exploitation with no or low care to the environment [29].

Finally the last issue is food distribution and availability and waste. The problem with food process, distribution and storage is that it is often difficult to ensure that food reaches those who need it in a timely and efficient manner. This problems causes a major food waste issue as it is considered that 30 % of food produces are wasted every year [30]. This can be caused by technical issues : Food may spoil or be wasted due to inadequate storage facilities, transportation difficulties, or other logistical challenges [31]. But this also can be due to social issues : There can be disparities in access to food, with some

countries, communities or individuals facing food deserts or limited access to fresh, nutritious food. These issues can lead to food insecurity and malnutrition, particularly for vulnerable populations such as low-income households, children, and seniors and the overall population in developing country. This inequality in food distribution leads to today absurd situation where 800 million people are undernourished while 800 million are considered in a state of pathological obesity [32].

All of those issues will require the adoption of more efficient farming practices that can produce more food with fewer resources. Thanks fully the digital revolution brought numerous tool to help make agriculture more sustainable. The domain of science and engineering that aim to bring new technologies to farmers is called Smart Farming or Precision agriculture. It is define as the usage of advanced technology to optimize agricultural production, increase yields and reduce waste and overall resources usage (mechanical and chemical) and finally make the overall process of food production more reliable. [33]. Smart Farming has already started to be implemented widely and it is consider that its usage will be even more important in the future. This lead to a new type of agriculture called 3.0 for the agriculture that already uses digital tools for automation and monitoring, and now even 4.0 with the introduction of advanced Information and Communication Technology (ICT) tools such as Artificial Intelligence (AI) or Internet of Things (IoT). Figure 1.4 resume the parallel between the level of agriculture and the level of traditional industry.

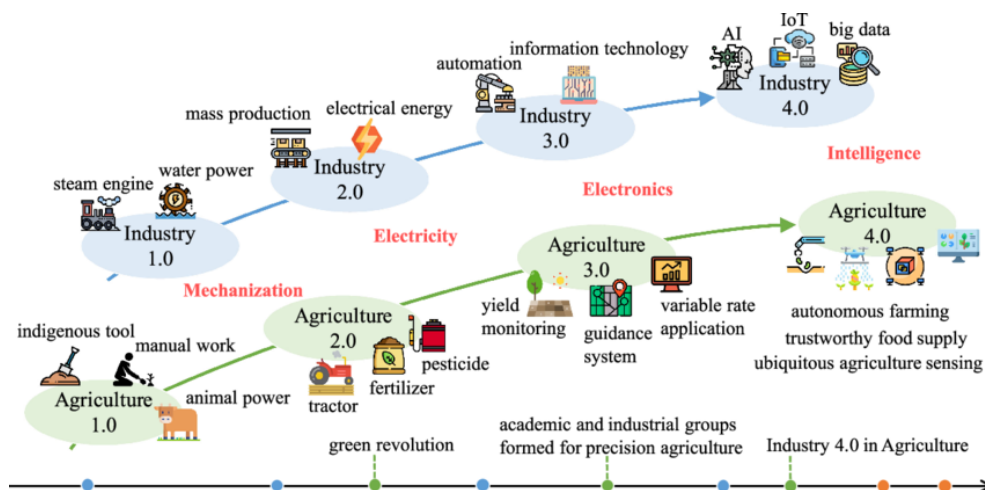


Figure 1.4: Development road-map of industrial revolutions and agricultural revolutions [34]

Concretely, smart farming involves integrating various technologies such as sensing technologies, data analytic, artificial intelligence and robotics. A general view of the smart farming processes is proposed in Figure 1.5.

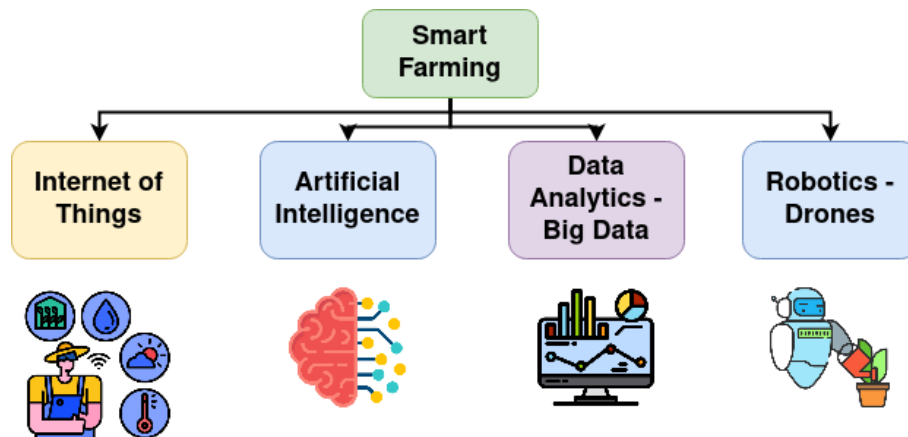


Figure 1.5: Smart Farming domains

All of these technology are generally grouped into Internet of Things (IoT) platforms to assist the farming process. The concept of the Internet of Things (IoT) refers to physical objects or collections of objects that are equipped with sensors, actuators, processing capabilities, software, and other technologies that communicate with other devices and systems over the internet. Usually IoT devices are grouped in IoT Platforms for specific usage. An IoT platform is software managing the connection and control of connected objects to collect, store, correlate, analyze and exploit their data and take the best decision to implement answers for various problem [35]. It is divided in three parts:

- **Perception:** Collection of data from various sources. Local sensors device, database, survey and human interaction systems.
- **Decision:** Analyse of such Data thanks to statics, analytics tools and AI
- **Action:** Sending command to actuators or proposing solution to human agent.

In the smart Farming context, a simple example of an IoT platform would be an irrigation system piloted by soil humidity sensors and is depicted in figure 1.6. The soil humidity sensors communicates wirelessly in what we call a Wireless Sensor Network (WSN). The IoT platform will collect the level of humidity in the soil, make analysis in comparison of future weather prediction (obtained through existing database on the internet), take a decision on the level of water to give to the crops and then finally send a precise activation command to a connected irrigation valve to apply only the minimum required amount of water, hence being precise and saving precious water resources.

Despite promising result and opportunity, the SF domain unfortunately encounters some limitations:

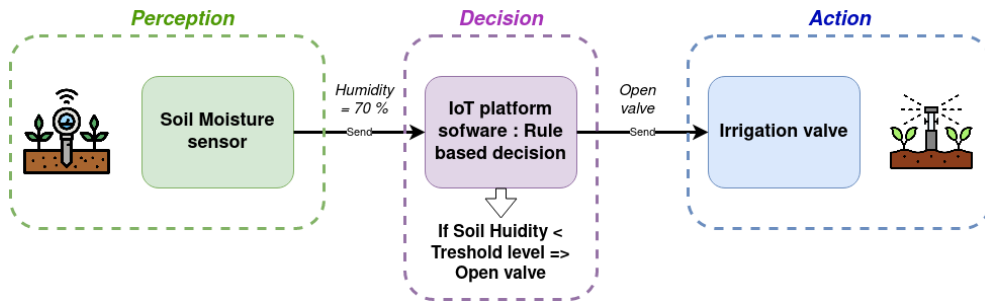


Figure 1.6: lot platform example

- **Technological** : The technologies used in SF are complex and need to be improved as explained by the authors of the survey in [36]. On the particular case of IoT platforms, which is the main topic of research of this thesis, the recent advancement in Embedded AI sensors offer promising application especially in SF. [37]. Figure 1.7 shows the rising number of publication concerning Edge computing technologies.

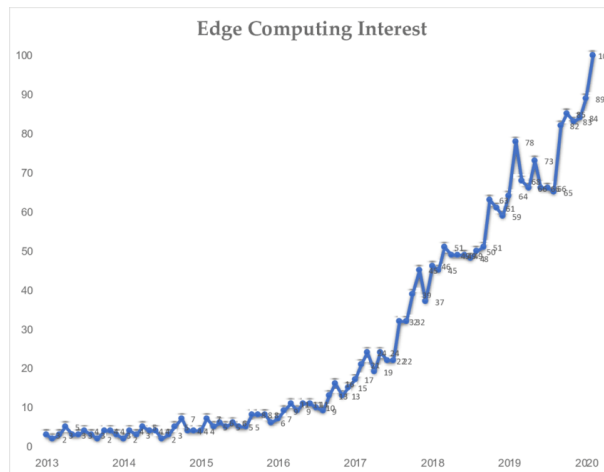


Figure 1.7: Edge computing interest (Google trends 2013-2020) [38]

Embedded AI is a new field of AI where Machine Learning algorithms are optimized in order to run on constrained hardware with low computational power and memory such as Microcontroller Unit (MCU) or embedded Central Processing Unit (CPU). The idea is to analyze the data directly where they are collected in order to reduce latency and increase privacy but also to reduce energy consumption as offloading data, especially heavy ones such as image and audio files, is a power-hungry procedure. Embedded AI is also referred to as Edge Computing in opposition of Cloud Computing. A simple comparison between cloud computing and edge computing is proposed in Figure 1.8 with an exam-

ple application where a camera sensor observe a field of strawberry. In the first case, cloud computing, the camera sensors take a picture of a plot of strawberry, sends the pictures through the network in a remote cloud server. The cloud server then apply a ML algorithm to this picture to infer the state of the plant. After this step, the cloud server can send the output to whoever needs to know (farmer, actuators (e.g. robot or irrigation)). In the second case, the ML algorithm is running directly on the sensors and it sends only the output in a form of a small and light text message through the network to whoever needs to know. Those

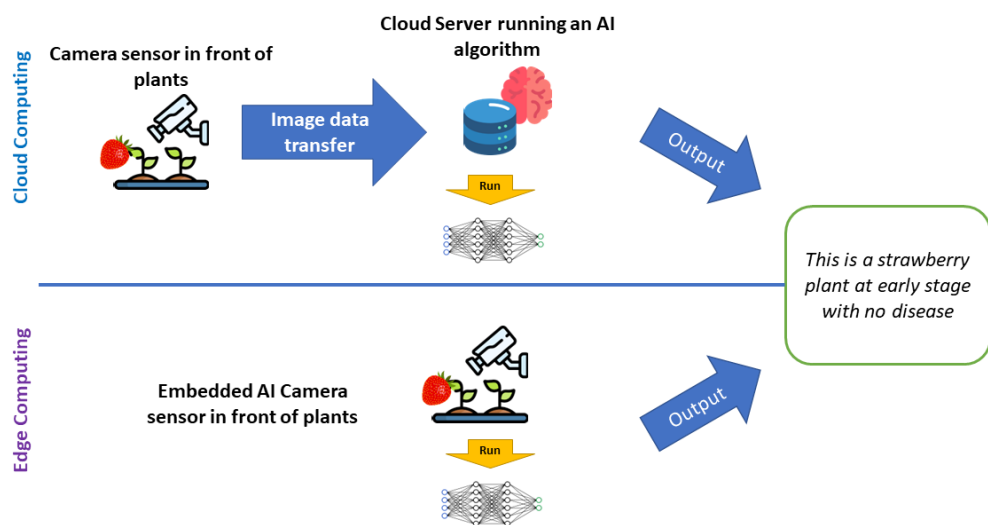


Figure 1.8: Cloud computing (top) vs Edge computing (bottom)

sensors offers a plethora of application for SF such as static camera sensors observing crops to monitor their growth and detect issues (e.g. disease, pest infestation, under development, water stress) or sound sensor to listen to livestock behavior and detect unhealthy individual for treatment (e.g. breath sensors for cows that detect stress). Unfortunately, they also brought new needs for agricultural Wireless Sensors Networks in term of Quality of Service (QoS) and mainly energy consumption[39][40] especially as their firmware need to be remotely updated and because they have to offload punctually some heavy data file (e.g. image or sound). As it will be later showcased, agricultural WSN mostly use Low Power Wide Area Network (LPWAN) technology to communicate [41]. Amongst those LPWAN, LoRa protocol is the main used and promising communication technology used in agricultural IoT platform for its long range performances and its capacity to create ad-

hoc network with unlicensed industrial frequencies [42]. Therefore the issue brought by Embedded AI sensors will be studied in the context of LoRa network. The intricate nature of IoT platforms is further complicated by these sensors due to their sophisticated management processes. It become more difficult for the system to represent the wide array of phenomena and variables the various sensors (smart or not) track, and ultimately, the diversity of devices (such as sensors, actuators, and routers) and agents (including plants and farmers) found on a farm. Numerous elements, ranging from sun exposure and water availability to natural competition and soil conditions, influence plant growth. In parallel, sensors have a plethora of parameters to consider, like their energy status, ambient temperature, and for those embedded with AI, their mean accuracy. For optimal decision-making, it's imperative to efficiently process the data collected and vigilantly monitor the health and status of each sensor. This necessitates the incorporation of AI methodologies such as knowledge engineering and expert systems to accurately represent our knowledge of both the sensors and the plants they oversee, allowing for well-informed decisions based on this comprehensive data.[43, 44]. Numerous methods already exist to do this such as ontology [45], but they have rarely been applied to agriculture and, to our knowledge, never to an agricultural IoT platform with embedded AI sensors [46, 47, 48]. Hence the strong focus of this thesis on this topic. Other technological issues will be briefly describe in the state of the art chapter of this report but the work conducted in this thesis was mainly oriented toward the issue related to the rise of Embedded AI in IoT platform using LoRa networking technology and the knowledge .

- **Social [49]:** SF research literature is mainly focused on technical implementations. However SF needs to also answer various social issues in order to encourage farmers to take the steps in changing their practices. Especially regarding data management and privacy as discussed by the authors in [50], or knowledge sharing with farmers as they will not use technology they don't understand and, therefore, won't be able to maintain as discussed in [51]. For example, one of the most significant reluctance of farmers is having to rely on technicians to maintain their trucks and equipment where they used to be able to do it on their own. Nowadays, the closed-source ecosystem of manufacturers forces farmers to hack their own tools and tractor to be able to repair them without paying fees to the manufacturer [52]. Therefore, the independence factor of farmers needs to be particularly addressed when developing smart farming solutions.

- **Economical [53]:** The gross margin in the farming industry are small. This is a key factor that SF tools need to take in account and it is closely linked to the social factor. Researcher must consider the economical cost of implementing them but also mainly maintaining them as it needs a specialised workforce to use them. For example an issue with the smart tractor proposed by one of the market leader manufacturer, John Deer's, has recently strike controversy as farmers can't repair their own truck due to proprietary software. When a malfunction occurs, tractor owners are often obligated to pay substantial repair costs to the respective company, a burden not easily offset by the farm's revenue [54]. Similarly, while agricultural robotics like automatic weeders and fruit pickers have shown potential (particularly in terms of sustainable development criteria), they face challenges in satisfying market demands due to their elevated costs and diminished return on investment compared to conventional human labor. Currently, only high-value agricultural products methods, such as viticulture, see a tangible benefit from these technological solutions [55]. In summary, smart agriculture must also be accompanied by favorable economic laws allowing its adoption, such as the prohibition of using cheaper but polluting methods or by imposing minimum wages for all agro-industry workers [49].
- **Scale [56]:** SF technologies are mostly applied to large scale farm that can have access to investment money to make the transition. In the opposite, Small Scale Farmer (SSF) are left out on the SF revolution as little to no research is conducted to offer them the tools to be more sustainable [57]. Small-scale agriculture consists of cultivating crops on a small surface of arable land, usually less than 2 ha according to most of the literature on the topic, and, often, yielding lower economic returns. This size number is subject to debate as a SSF in Latin America is not the same as one in Australia for example but also that surface itself is not enough of an indicator, as we should also consider the productivity, the income or the quantity of livestock of a farm [58]. The study proposed by the authors in [57], shows that research is not on the right track as 95 % of the 100 000 articles published on the ways to end hunger were of no help for small-scale farmer. However, small scale farming represents the vast majority of the farm in the world, over 80 %, as shown in Figure 1.9 :

Due to its poor yield, its sensitivity to the climate and the intense work labour, most of the farmer around the world operating in SSF are from the poorer countries of the planet [60]. Nowadays, SSF are the subject of huge interest as their development could help the population of country under development to be more self sufficient and resilient to hunger, and to offer economical revenue to population [61]. But an-

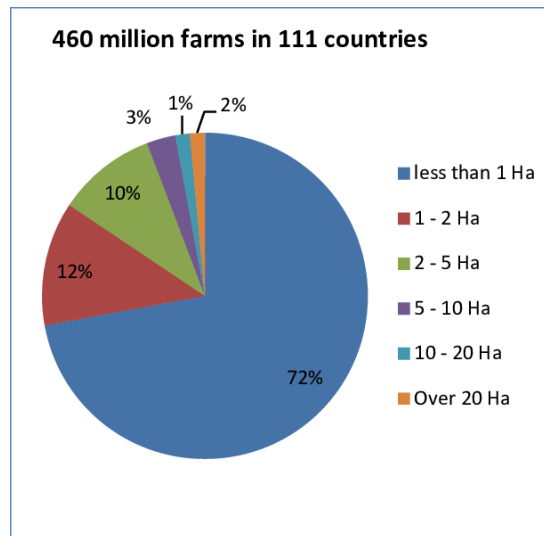


Figure 1.9: Share of farms worldwide, by land size class. [59]

other huge interest for those SSF is about their capability to be more sustainable and resilient than large-scale one [62, 58]. For those reason, research are funded even in Europe to find new ways to allow people to go back to small scale farming [63]. The problem of small scale farm is mostly the cost of the workforce. To reduce it two solutions can be used. First making the economical revenue more profitable by applying dedicated policies. Second reducing the needed workforce by using technology. This is on this second point that I pursued my research, even if the social and economical dimension of the problem should still be remembered during technical research in order to offer a useful solution to tackle with the issues of farmer.

- **Application domain:** The last limitation is the type of agriculture SF is applied to [64]. As for now most of the project are used on conventional agriculture methods such as mono-cropping with fertilizer usage for example. Despite offering gain in productivity, those methods are not sustainable in the long term as explain before in this report. Hoppefully, true sustainable methods exist such as agroecology [65]. Agroecology is a scientific discipline that focuses on the study of ecological processes that operate in agricultural systems, with the aim of developing sustainable and resilient food systems. It emphasizes the interdependence of the social, economic, and ecological dimensions of agriculture, and seeks to promote the use of ecological principles and practices in agriculture. Agroecology emphasizes the importance of the local ecosystem, biodiversity, soil health, and the use of renewable resources, and aims to reduce the dependence on external inputs such as

chemical fertilizers and pesticides. It also seeks to promote social justice and equity in the distribution of resources and benefits within the food system [66]. The philosophy of agroecology is simple : Work with nature and not against. This mean adapting the crops to grow to local climate by artificial selection, using other biological agent to counter undesired effect (e.g. introducing predatory species to a know unwanted insect) or simply performing multi-cropping in a same field to have beneficial interaction amongst plants. The best example of agroecology is a method called three sisters that was, and is still use by the indigenous people of south and north america [67]. It uses a combination of squash plant, maize (corn), and climbing beans in a technique known as companion planting [68]. The maize and beans are often planted together in mounds formed by hilling soil around the base of the plants each year; squash is typically planted between the mounds. The cornstalk serves as a trellis for climbing beans, the beans fix nitrogen in their root nodules and stabilize the maize in high winds, and the wide leaves of the squash plant shade the ground, keeping the soil moist and helping prevent the establishment of weeds. A representation of this process can be found inf Figure 1.10. Association like this between plants exist in a wide variety and have various objectives. The multiplicity of such association make knowledge engineering an interesting tools to represent them in regard of their independent biological characteristics.

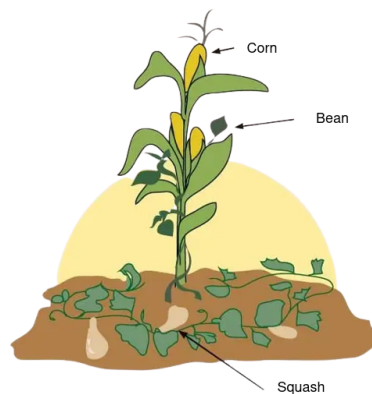


Figure 1.10: Three sisters method representation

In light of the agricultural context and the opportunities and limitations of smart farming, this thesis research has been directed towards implementing an IoT platform for small-scale agroecology. There is a significant emphasis on networking challenges associated with wireless embedded-AI sensor networks and the knowledge engineering required by the decision-making processes related to sensor management and agroecological principles

1.2 . Problem statement

Small-scale agroecology can be a viable model for sustainable food production in the future. Unfortunately smart-farming tools that address the specific issues encountered by those type of farm are scarce and under-developed compared to other type of agriculture (conventional agriculture or indoor farming). Hence the main problematic of this thesis :

"How can we improve farmer's ability to use smart-farming tools to implement sustainable agroecological practices?."

As stated above, IoT platform are powerful tools to gather data and interpret them in order to take the best decision regarding certain rules. This capacity to perceive information in a farm through sensors is even more efficient with the rise of new smart sensors performing AI directly at the Edge on the sensors hardware. This thesis will therefore focus on the implementation of an IoT platform with an heterogeneous WSN (A network composed of Smart sensors and classic sensors), to help the farmers follow agroecological principle that rely fundamentally on the observation and interpretation of the biotope and biocenose behavior of the farm. To do so we ask the question :

How can IoT platform be an answer to manage agroecological farms ?

To do so, we ask the question of the maintenance of the wireless sensors network as it should be minimal to avoid farmer's intervention. The main maintenance task that farmers have to perform with sensors is Battery management. Agricultural WSNs mainly use Low Power Wide Area Network (LPWAN) technologies, especially LoRa telecommunication protocol to communicates in order to save energy and doing so, extending sensor's battery lifetime drastically. Unfortunately those network have a really low data-rate. Therefore the rise of the usage of new smart sensor needs to be addressed in regard of LoRa limitations and advantages. The two main need for smart sensor, beside the communication of the results, is the ability to be remote updated and the ability to offload punctually an heavy file (e.g. image or file) for re-training the model in the cloud.

How can smart sensors be used in LoRa agroecological WSN ?

Finally, we need to represent the knowledge of agroecology and the knowledge on sensors network behavior (with a strong focus on Embedded AI sensors) in a novel Knowledge base to propose the best decision to the farmer regarding sensor placement, sensors maintenance (energy saving procedure & Firmware update management), crops layout, action determination (irrigation, harvest, treatment). For that we will be using Knowledge-engineering tools such as Ontology. In this context, we finally ask the question :

"How can we merge the information gathered by the sensors and take the best decision based on their data to implement agroecology and manage the sensors network maintenance?"

To conclude, the proposed IoT platform should be able to merge data from the variety of implemented sensors in order to take the best decision regarding farm management but also WSN management with the help of a knowledge base. The augmented IoT platform compare to the one in figure 1.6 is shown in figure 1.11. It should be able to help sensors management to allow farmer with limited knowledge of IoT platform to implement easily such tools. Despite a strong focus of this thesis on the locally deployed WSN, it is important to note that the platform also need other source of information to output more precise decision. Those external sources are remote sensing techniques (satellite imagery) and farmer's observation on the field. Therefore those data source should also be taken into account in the process of decision making.

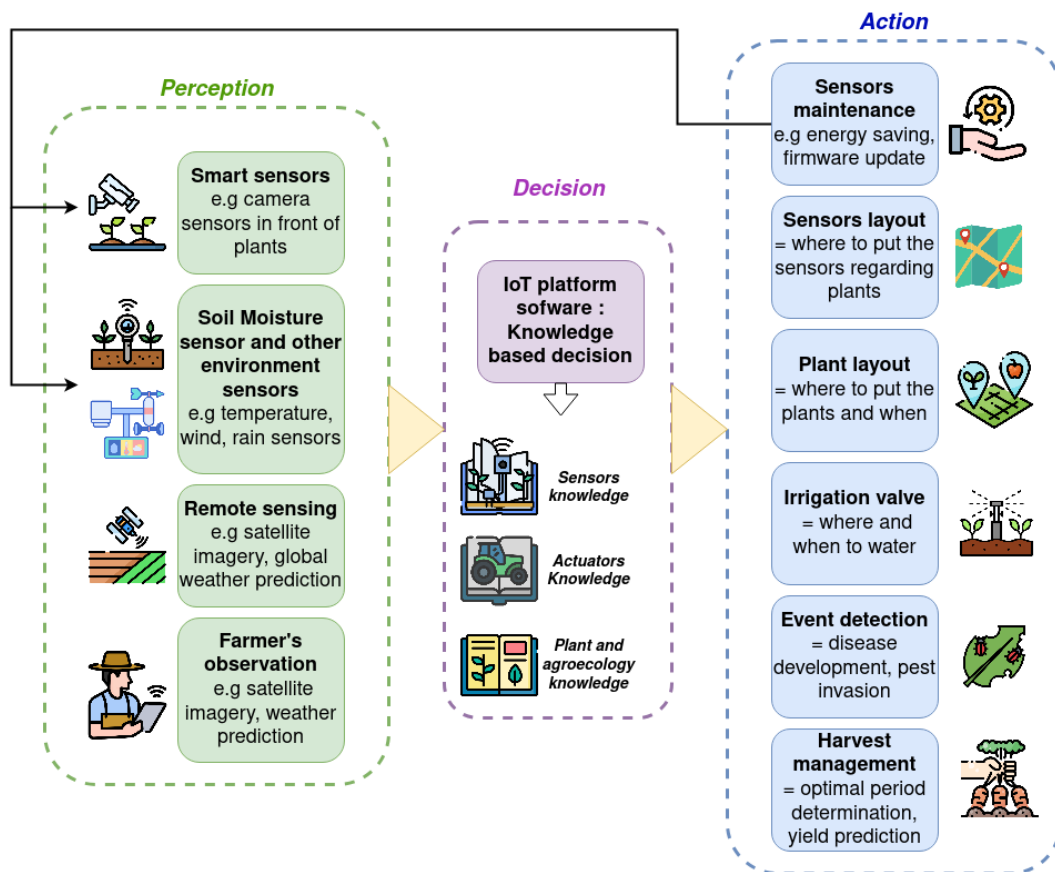


Figure 1.11: Overall view of the IoT platform

1.3 . Methodology and Contribution

The goal of this work is to design an efficient IoT platform for small-scale agroecology. To do so, as stated above, we need to consider the specificities of the introduction of smart sensor in wireless sensor network using LoRa technology and in the meantime treat the perceived data through a knowledge base regarding plant biology and sensors characteristics. The different parts of this work can be represented as a methodology in Figure 1.12. This methodology outlines the different phases of this work, their corresponding chapters, and the peer-reviewed contribution published during the thesis.

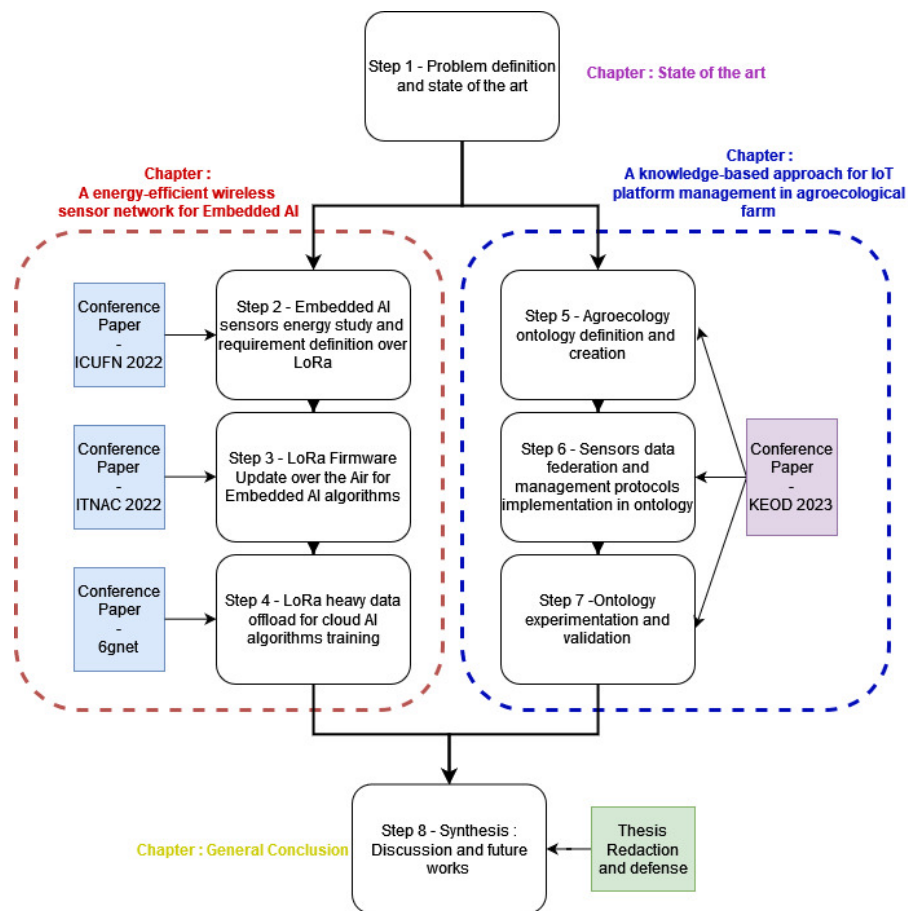


Figure 1.12: Methodology and contribution overall view

1. The first step and contribution is a standard approach of scientific literature exploration in the concerned specific domains : Smart Farming, IoT platform for agriculture with strong focus on Wireless Sensor Networks, embedded AI devices applications and knowledge-based decision systems.

2. The second, third and fourth step are about the specific needs of Embedded AI and smart sensors in LoRa and are grouped in one chapter. The second step contribution is a proof of energy-efficiency for system using Embedded AI and LoRa compared to cloud computing.
3. The third step is about developing a Firmware Update Over the Air protocol in LoRa for Embedded AI devices performing TinyML.
4. The fourth step is about developing an heavy data-offload protocol for LoRa for remote model training.
5. The fifth, sixth and seventh step are about the setup of the knowledge-base that will act as a way to store the perceived element and process them through a set of logical rules for a decision-making purpose. The fifth step is about representing the knowledge about plants and agroecology principles.
6. The sixth step is about representing the knowledge about sensors (especially smart sensors and their maintenance procedure) and WSN principles with a strong focus of storing the measured data.
7. The seventh step is about experimenting the created Knowledge-Base on use cases to get results. This is done in a simulated environment and by exploring different scenarios.
8. The eighth and last step is about a synthesis in which results are discussed and future works are presented.

Overall, this thesis propose approaches for the implementation of an IoT platform for agroecology in the form of a cloud platform where the farmers can enter his information about his farm location and types of plants he wish to grow and obtain a crops and sensors layout from the machine. Once he implemented this crops and sensors layout in the farm, the platform manage the sensors maintenance (meaning the battery usage, the FUOTA and data offload procedure for the Smart sensors) and the crops development with a knowledge base ontology. This IoT platform is presented in figure 1.13.

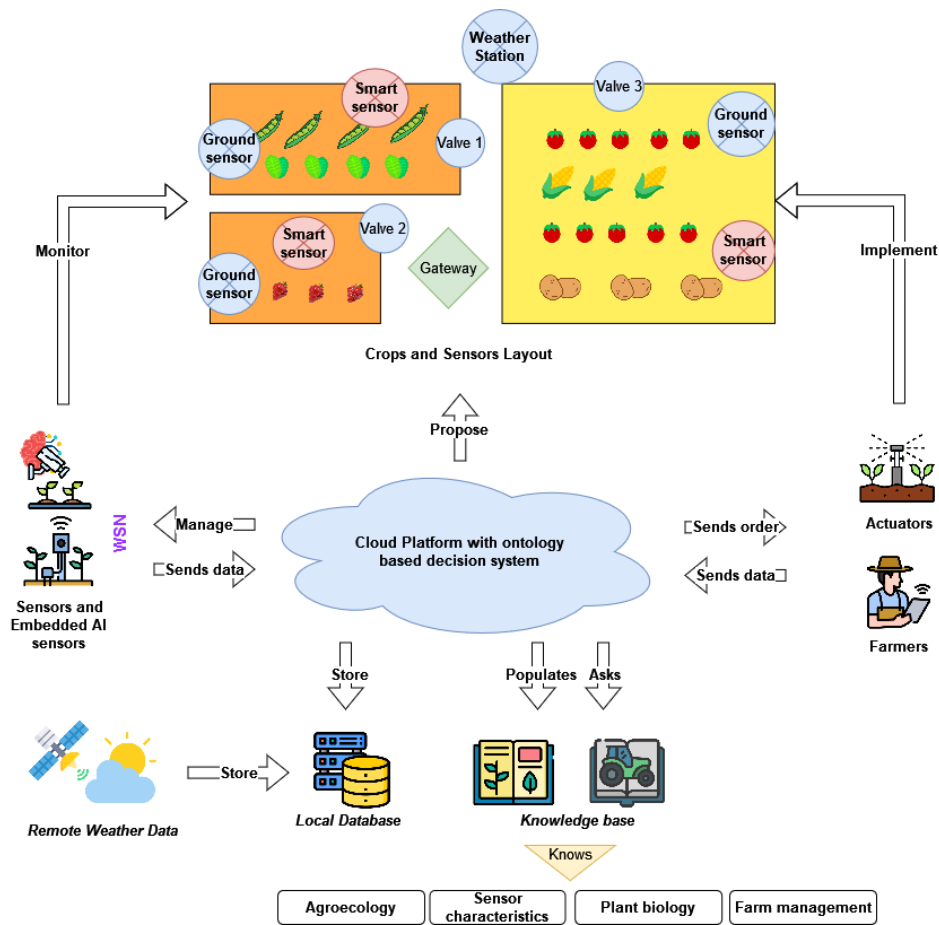


Figure 1.13: IoT platform specific view

1.4 . Report Outline

In this report, Chapter 2 examines the state of the art in relation to IoT platforms in agriculture. In this context, the study focuses on two specific aspects of the IoT platform. Firstly, it addresses the utilization of embedded-AI sensors and their communication requirements within wireless sensor networks, emphasizing LPWAN and LoRa technology. Secondly, it delves into the storage, federation, and interpretation of sensor data for Agroecology. This chapter thus provides a review of WSN, Smart Sensors, and knowledge-based systems. Their current limitations and potentials are highlighted in the context of Smart Farming (SF). The chapter concludes with a summary of the state of the art and an overview of the proposed system. It's worth mentioning for the reader that this chapter is quite extensive. The objective was to offer a thorough overview and highlight the research efforts necessary to create a comprehensive knowledge base. Some of the concepts explained are aimed at individuals without prior experience or knowledge in the various domain

and may thus be bypassed by well-versed readers.

Chapter 3 covers the topic of LoRa-based agricultural WSN with Embedded-AI sensors. It introduces a particular use-case scenario that highlights the potential application of such sensors. In the process, it thoroughly examines the limitations inherent to these sensors within the realm of Low Power Wide Area Network (LPWAN) communication, with a specific focus on the LoRa protocol. These limitations include energy consumption, Firmware Update Over the Air (FUOTA), and the challenge of handling substantial data offloading. Afterward, we outline the software and hardware architecture of the embedded-AI algorithms we developed using Tiny Machine Learning techniques (TinyML) for our experimentation. We then present two novel protocols for Smart Sensors: one for FUOTA and another for efficient heavy data offload. Finally, we simulate our sensor system's behavior in various scenarios to explore its capabilities.

Chapter 4 focuses on the Knowledge-Engineering approach within the thesis. Firstly, it will define the concept of knowledge engineering and its realization through ontology development. Subsequently, it will showcase the knowledge-base constructed for our IoT platform. The latter will introduce the ontology's constituent classes, objects, and properties regarding plants and IoT devices. Moreover, it will outline a set of rules employed by the reasoner tools for the inference process. Beyond mere data storage, the system must also effectively handle diverse actions related to perceived data and implemented knowledge. Ultimately, we obtained an ontology tailored to the management of a small-scale agroecological IoT platform and delve into its architectural aspects. Chapter 4 also introduces various use-case scenarios to validate the behavior of the IoT platform's knowledge-base. The model underwent testing within a simulated environment to verify the suitability of the approach for both sensor and plant management.

Finally Chapter 5 will gather the various conclusion of the conducted research and propose future paths to explore in order to improve our system. This chapter will also showcase the various achievements attained during my PhD journey and highlight prospects for future career opportunities.

2 - State of the art

2.1 . Introduction

Smart farming is a scientific domain aiming to incorporate digital tools into the agriculture industry to enhance productivity, reduce labor, and improve food quality. SF is also called Precision Agriculture (PA) and can also be defined as the use of technology to improve the ratio between agricultural output (usually food) and agricultural input (land, energy, water, fertilisers, pesticides, etc.). As outlined in the introduction of this thesis, Smart Farming, could significantly contribute to shifting agriculture towards more sustainable practices while ensuring global food security [33]. SF technology spans various computer science fields, including Artificial Intelligence, Data Science, and Robotics. Through cloud-base AI analysis, farmers can better comprehend climate patterns and changes, facilitating informed decisions about crop selection and resource allocation (e.g., water, chemicals). In the meantime, actuators such as Robots can effectively replace humans in challenging tasks like fruit harvesting and weed removal or simply in automation task like valves for irrigation management. The different components of Smart Farming are illustrated in Figure 2.1.

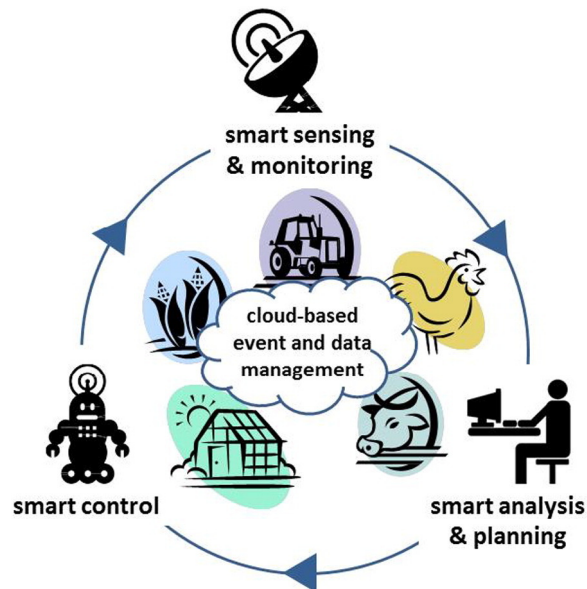


Figure 2.1: The cyber-physical management cycle of Smart Farming enhanced by cloud-based event and data management [69]

However, as mentioned in the introductory chapter of this thesis, SF faces

limitations in social, economic, application domain, scale, and technological aspects. This thesis concentrates on the challenges associated with utilizing Internet of Things (IoT) platforms within the SF context. Previous studies have explored IoT platform deployment in farming environments, offering substantial potential to enhance agricultural sustainability. IoT platforms must ensure communication among diverse networked devices, often wirelessly. This can be complex, given the diverse requirements of these devices, particularly due to the growing integration of embedded AI. Also Upon gathering data, IoT platforms must also make optimal decisions using acquired knowledge [35].

Due to its multidisciplinary nature, this problem entails the convergence of multiple subjects, necessitating a comprehensive scientific review of seemingly disparate areas. This chapter compiles information on the current state of the art and direction of Smart Farming technology, particularly in relation to Internet of Things applications. The focus is on wireless communication, embedded Artificial Intelligence, and knowledge-based decision-making processes. This section seeks to provide readers with a comprehensive overview of these technologies, their common applications in agricultural scenarios, their limitations, and potential areas for improvement. Other detailed technical reviews will be provided in Chapters 3 and 4, focusing on the specific applications implemented in our IoT platform.

2.2 . IoT platform

2.2.1 . Definition

An Internet of Things (IoT) platform refers to an integrated technological infrastructure designed to facilitate the seamless connectivity, data exchange, management, and control of a diverse array of physical objects or "things" through the Internet. These "things" encompass a wide spectrum of devices, sensors, actuators, and equipment that can collect, transmit, and receive data. IoT platforms serve as foundational frameworks that enable the interoperability and communication among these interconnected devices, allowing for the aggregation, analysis, and utilization of data for various applications. In other terms, it is a computer system capable of making autonomous action in the world based on its perception of it [35].

The concept of "Internet of Things" was initially coined by Kevin Ashton in 1999 [70]. In a presentation he made, Ashton highlighted the potential advantages of employing RFID technology for managing products. By outfitting items with specific devices, these products could "communicate" pertinent details (such as their condition, traceability, etc.). This would enable products

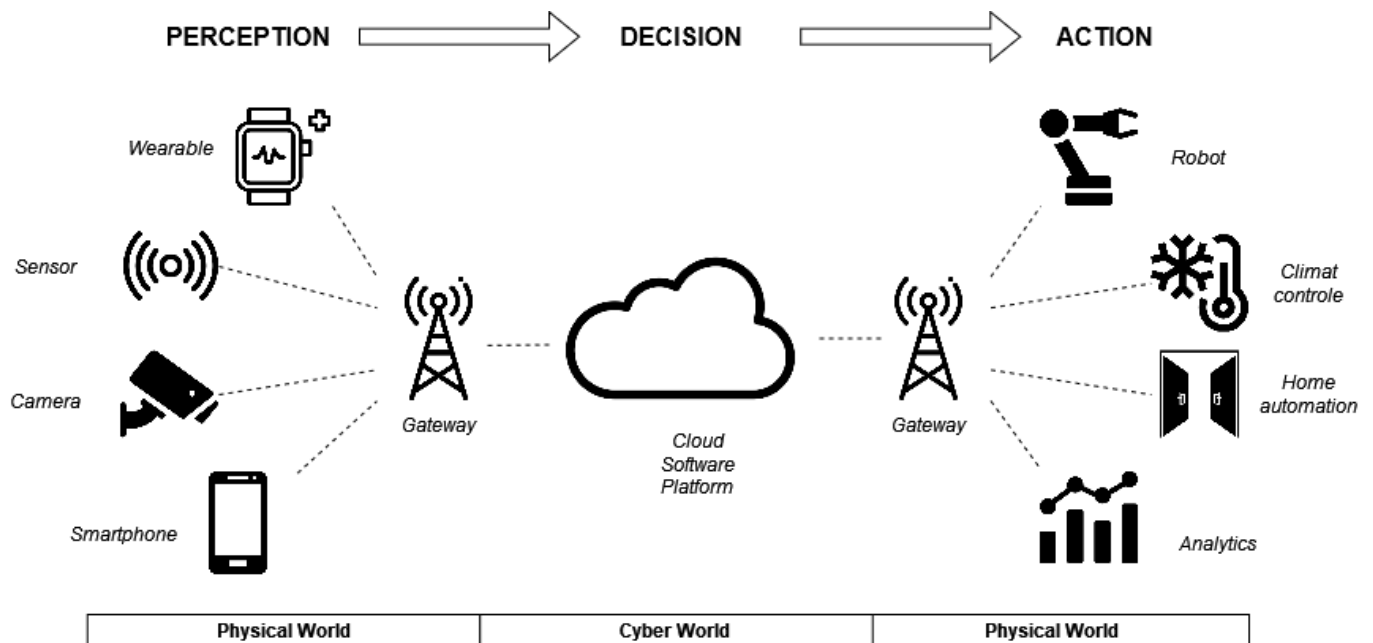


Figure 2.2: IoT platform model

and individuals equipped with such "things" to share insights about their condition and environment in a vastly improved manner.

It is composed of three main components: Perception, Decision, and Action [71]. The perception module is responsible for retrieving data, either from a cloud database or through human interaction, and directly from the physical world using sensors. The decision module is software tasked with interpreting the collected data through data analytics or AI. Finally, the action module concerns the implementation of the decisions made. This can take the form of other physical devices known as actuators or robots, or in the form of reports communicated to humans to help them make a decision or to give them guidelines. Each component of an IoT platform communicates via specific transmission protocols regarding the specifics of the device [72]. Communication can be wired or wireless. The general model of an IoT platform is depicted in Figure 2.2.

For a deeper level of understanding, we present an illustrative example of an IoT platform in the context of Agriculture in Figure 2.3. The primary goal of this platform is to monitor stress levels in livestock, enabling the adjustment of the type and quantity of food they receive. This adjustment is crucial as food significantly influences stress levels [73]. Prior to IoT implementation, farmers had to manually observe signs of stress in individual animals. This task was challenging, especially for a large cattle herd, relying solely on visual

cues of animals behavior. When an individual exhibited signs of stress, the farmer had to make approximate adjustments to their food regimen. However, with IoT technology, a sensor can be directly attached to each animal, allowing for the precise monitoring of stress levels using physiological sensors such as heart rate and breath sensors. Additionally, strategically placed cameras in the fields can analyze cow behavior through AI algorithms. Cameras and sensors need to communicate over the network to share their data with the cloud-based platform. Communication can either be wired, as seen with the cameras, or wireless, as is the case with the sensors attached to cows. Based on the collected data, the system can then precisely fine-tune the quantity and type of food provided to each individual. Furthermore, this data-driven system generates analytical insights for farmers. It can even automate the process of ordering animal food based on the actual needs of the cattle herd. Thanks to this system, one farmer can now monitor the stress level in a cattle herd of thousands of beasts easily and implement basic solution to diminish it.

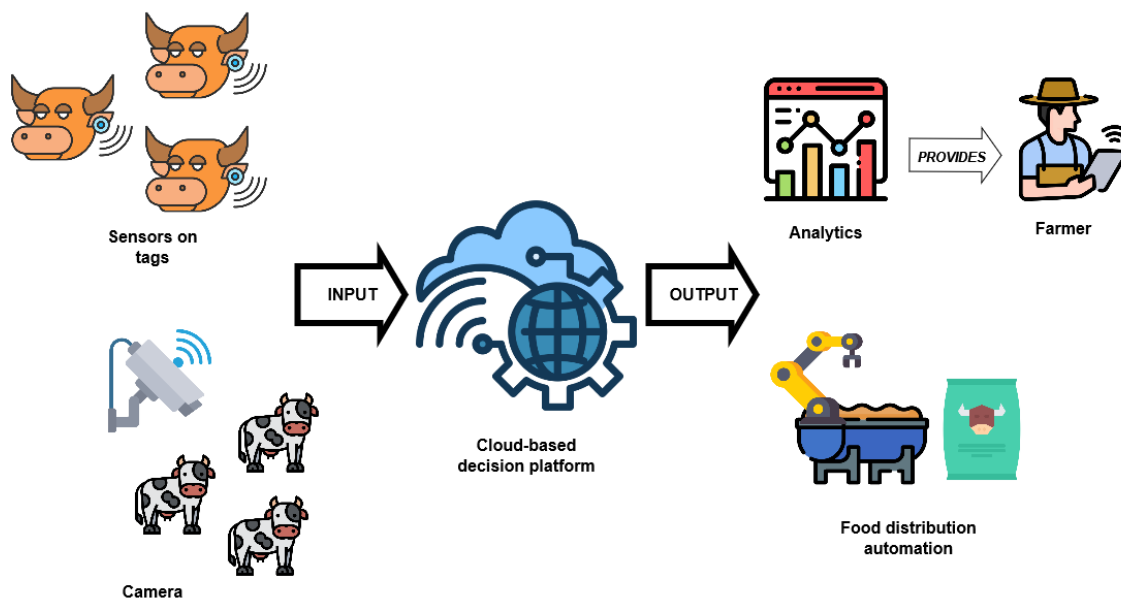


Figure 2.3: IoT platform for livestock stress management

The previous example is straightforward, but an IoT platform can become significantly more intricate due to the diverse array of data sources available for the process of decision-making and the technology chosen for it. These platforms might even identify novel stress patterns in livestock that elude human observation. In the broader scope, IoT platforms can also facilitate predictive maintenance in industrial settings [74]. In our example, this involves averting stress in livestock by detecting early signs of abnormal behavior. This

can be achieved notably through the utilization of a concept known as the "digital twin." The concept of a digital twin in IoT involves creating a virtual representation of a physical object or system. This virtual counterpart mirrors the real-world object, collecting and utilizing data from sensors and other sources to simulate its behavior, performance, and condition. This enables real-time monitoring, analysis, and optimization, helping to enhance decision-making, predictive maintenance, and overall efficiency [75].

When considering the technical architecture of an IoT system, arriving at a universally applicable design is a complex endeavor. The primary complexity arises from the inherent divergence among potential applications, each contingent upon a diverse array of variables and distinct design specifications [76]. In order to be efficient, an IoT platform should be:

- **Scalable:** Ensuring that as the number of devices and services grows, their performance remains unaffected.
- **Interoperable:** Enabling devices from various manufacturers to work together towards shared objectives;
- **Distributive:** Facilitating the establishment of a distributed framework where data, collected from multiple sources, is processed in a decentralized manner by different entities
- **Secure:** Preventing any unauthorized access
- **Ressource constrained:** Most objects typically possess limited computing capabilities

The most common architecture found in the literature, as explained by the authors in [77], is the Three Layer Architecture. This architecture is presented in Figure 2.4 and others common architecture are described by the authors in [78]. It is composed of the following layers:

1. **Perception/Action:** It symbolizes the physical layer of objects that engage with the surrounding environment by gathering data and/or performing tasks. This layer encompasses objects equipped with computing and communication features, enabling them to interact with the external world. Those "Things" are usually equipped with the following essential properties:
 - *Communication:* Objects can communicate with others and with different resources over the network
 - *Identification:* Objects are uniquely identified

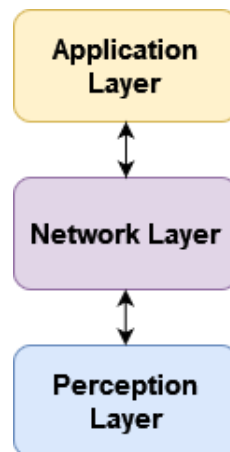


Figure 2.4: 3 layers architecture

- *Sensing*: Objects can perceive physical phenomenon in their environment.
- *Actuating*: Objects can interact with their environment.
- *Computing*: Objects can process information.
- *Localization*: Objects can be mapped.
- *User Interface*: Objects can be interact with by human being.

Various technologies can be used to implement those properties[79]. In table 2.1, we present example technologies for the different properties except the communication that will be explore more in depth on the network part.

2. **Network**: The network layer's responsibility involves conveying the data collected at the perception level to the application layer or transmitting data from the application layer to the actuators or target services. This encompasses all the technologies and protocols essential for establishing this connection. It's important not to mistake this network layer with the one in the ISO/OSI model, which solely directs data through the network along the optimal route [80]. A wide variety of protocols are available for utilization in IoT. Table 2.2 illustrates several of the frequently employed protocols, categorized based on the ISO/OSI model. Each protocol boasts its own advantages and disadvantages, necessitating careful consideration of their applicability according to specific use cases. We will explore in more depth the various protocols in the next section about Wireless Sensors Networks 2.3.
3. **Application**: The application layer encompasses all the software required to deliver a particular service. At this level, data from the pre-

Properties	Used Technologies
Identification	Electronic Product Code (EPC), Ubiquitous Code (uCode), Quick Response (QR), Original Equipment Manufacturer (OEM) Serial number, etc.
Sensing	Optical Sensors, Electromagnetic sensors, Remote sensors (Satellite, Aerial imagery), Local sensors (temperature sensor, humidity sensor), cameras, microphones etc.
Actuating	Mechatronics and Robotics, Lights, water control, climate control systems (Air, Heat), domotics etc.
Computing	Microcontrollers (Arduino,STM32 etc.), System on Chip (SoC) and single board computer (Raspberry Pi, Nvidia Jetson), Field Programmable Gate Array (FPGA) and other Programmable Logic Controller (PLC)
Localization	Global Position System (GPS), Galileo, Network triangulation, etc.
User Interface	Displays, keyboards, buttons etc.

Table 2.1: Usual technologies used in the IoT domain

ceding layers are collected, consolidated, screened, and processed using databases, analytical tools, AI, and other software. It is important to note that the computing process of the application layer can happen at different location depending on the IoT platform. If the computation process happens directly on the devices, we talk about edge computing, if it happens in remote distant server accessible through the Internet we talk about Cloud Computing and for the computations happening in between (at the gateway, on servers present on the local network physically) we talk about fog computing. We will come back on this notions later on this thesis in section 2.4.

To conclude, the Internet of Things (IoT) represents a new technological paradigm with significant and forthcoming applications in three main domain: Consumer IoT (e.g. smartphones, smart car, smart watch, etc.), Commercial IoT (IoT Healthcare, Smart City, etc.) and Industrial IoT or IIOT (manufacturing, aeronautics, agriculture, etc.). Consequently, it is regarded as one of the most rapidly expanding markets of this decade. Projections suggest that the

ISO/OSI Model layer	Network communication protocol
Application	CoAP, MQTT, AMQP, XMPP, DSS; Service Discovery: mDMS, DNS-SD, SSDP; Security: TLS, DTLS
Transport	TCP/UDP
Network	Adressing: IPv4/IPv6 Routing: RPL, CORPL, CARP, etc
Adaption	6LoWPAN, 6TiSCH, 6Lo, etc.
Data Link	IEEE 802.15.4 (Zigbee, etc.), IEEE 802.15.1 (Bluetooth), LPWAN (LoRaWAN, etc.), RDFID, NFC, IEEE 802.11 (Wifi), IEEE 802.3 (Ethernet), IEEE 1901 (PLC), I2C, SPI etc.

Table 2.2: Main network protocols according to OSI/ISO Model used in the IoT field

number of IoT devices will surge from 7.6 billion in 2019 to 24.1 billion by 2030, showcasing a compound annual growth rate (CAGR) of 11%, as indicated by recent research [81, 82].

In terms of revenue, the collective IoT market was valued at USD 465 billion in 2019, a value that is expected to escalate to USD 1.5 trillion by 2030. The majority of expenditure, around 66%, will be allocated to services such as connectivity, while the remaining portion will be attributed to hardware, encompassing dedicated IoT devices, modules, and gateways.

The Agricultural IIoT sector is projected to encompass 4% of this market, equating to over USD 60 billion by 2030. Those statistics about the IoT market can be found in Figure 2.5.

2.2.2 . Agricultural Application

The realm of IoT applications spans across a multitude of industries, encompassing manufacturing, healthcare, transportation, energy, retail, smart cities, logistics, and home automation. In this context, our attention will be directed towards its specific application in agriculture. The research on farming industry is well-funded and in the middle of its most important transition toward sustainability. As a result, a multitude of IoT initiatives have emerged within this sphere, driven by research groups, companies, and individuals alike. Our aim is not to conduct an exhaustive examination of all IoT applications within the agricultural domain, as comprehensive surveys on this subject have already been conducted by the author in [83, 84, 85, 86]. Rather, our objective is to provide the reader with a concise overview of existing projects, facilitating a better understanding of the extensive array of potential applications within the realm of smart farming while outlining common limitations.

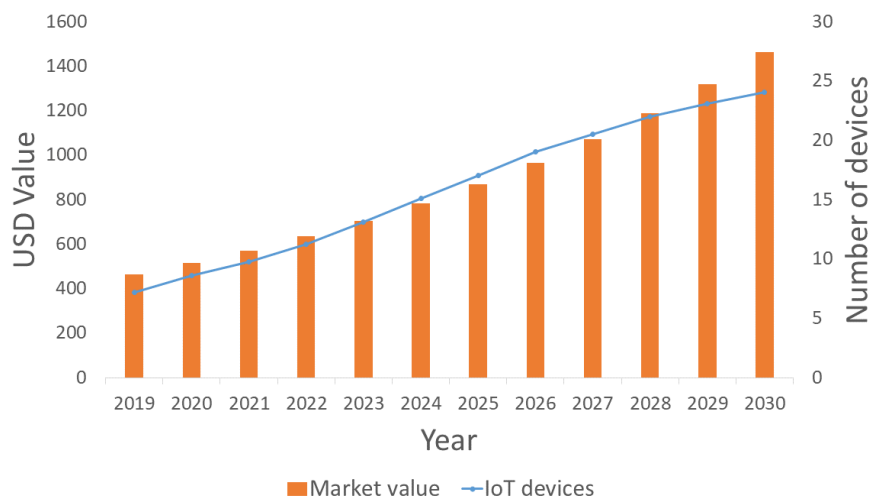


Figure 2.5: The internet of Things Market 2019-2030 (number of devices in billions and USD value in millions)

We listed and compared some project in different application domains:

1. *Irrigation management:* The most prevalent application in agriculture is the control of irrigation. Reducing water supply to crops results in slower growth and diminished calcium absorption. Frequent irrigation leads to root decay and water wastage. As outlined by the UN Convention to Combat Desertification (UNCCD), around 168 countries are projected to confront desertification by 2030, with nearly 50% of the global population residing in regions experiencing significant water scarcity [87]. Consequently, it becomes crucial to reduce agriculture’s dependency on water resources through enhanced precision in its utilization. To do so, farmers install drip irrigation systems to easily deliver water to each crop [88]. Those drip irrigation system are now often managed through IoT, therefore multiple research project have been conducted and the authors in [89, 90, 91, 92, 93, 94] proposed good surveys of those projects. Based on sensors (local, like electronic soil moisture sensor, and remote like satellite imagery), those platform can determine and send exact amount of water to each type of crops. One of the most basic example of such project have been implemented by the author in [95], where they use arduino microcontroller, a capacitive soil moisture sensor and a cloud-based decision platform to manipulate an irrigation valve. More advanced application, like the one proposed by the author in , use AI and fuzzy logic algorithm to determine the water-need [96]. From our literature review we determined the following parameters to take in account for plant irrigation: Soil Moisture,

Soil temperature, Soil type and structure, Air Temperature, Air humidity, Wind, Rain, Solar exposition, Satellite imagery, Weather forecast, type of plants, calendar, localization.

2. *Soil management*: Crops' growth heavily depends on soil fertility. Soil fertility is determined by the quantity and availability of specific nutrients in the soil [97], as well as its structure [98]. Nutrients are minerals present in the soil that are essential for crop growth. A multitude of nutrients exist in various types (e.g., macro and micro), with Nitrogen (N), Phosphorus (P), and Potassium (K) being the most critical ones [99]. Quantifying nutrient quantities in soil involves complex chemical analyses often requiring advanced lab equipment, such as spectrometers [100].

Soil structure involves arranging soil particles (e.g., sand, silt, clay, organic matter) into porous compounds, known as aggregates. These aggregates' arrangement, separation by pores and cracks, and composition influence soil structure. Soil structure also dictates its pH [94]. Just as determining these characteristics often demands advanced laboratory tools, their monitoring is vital for informed decisions regarding fertilizer application, crop selection, and irrigation methods. Hence, research has explored IoT platforms for real-time soil condition monitoring.

Regarding nutrients, [101] and [102] proposed a Wireless-Sensor Network utilizing an electrochemical sensor to monitor N, P, and K levels. This system, after nutrient level determination, refers itself to databases of soil nutrient requirements for different crops at different growth stages, in order to suggest fertilizers application to attain ideal nutrient values. Authors in [103] employed a more precise colorimetry sensor for similar outcomes. Other researchers have introduced an X-Ray fluorescence approach based on a camera, enhancing nutrient assessment accuracy. This project uses AI to identify significant characteristics in camera images. AI also determines fertilizer levels and suggests efficient crop rotations, aiding soil fertility recovery between different cultures [104, 105].

For pH measurement, real-time monitoring is also important. Researchers have developed IoT-based monitoring schemes, as seen in the works of [106] and [107]. In contrast, soil structure monitoring in real time is less essential, as it generally remains constant over time [98]. However, soil compaction, a significant issue linked to agriculture, affects soil structure. Highly compacted soil reduces fertility, hindering root development and nutrient/water uptake [10]. To detect and prevent soil compaction, various sensors, including mechanical ones, and IoT systems

have been proposed [108].

In conclusion, IoT proves to be an effective tool in preventing soil degradation, encompassing erosion, alkalization, acidification, salinization, and pollution [109].

3. *Crop Diseases and Pest Control:*

Usually, crop disease and pest detection are carried out by human operators since the primary information collected is visual. Farmers roam their fields, observe pests or diseases, and then implement corresponding countermeasures. These countermeasures can be categorized as organic, chemical, or mechanical. Organic approaches involve the use of sustainable substances to protect plants from threats. Conversely, chemical methods use products that may be more effective but are not sustainable. Mechanical countermeasures typically involve removing affected plants and surrounding vegetation to stop the spread of disease and pests [110].

Regrettably, the agricultural workforce has significantly diminished over the past century. As a result, manual monitoring of all crops by farmers is no longer feasible [111]. Hence, current methods often involve preemptively applying chemical countermeasures before pests or diseases have a chance to develop, with the aim of preventing their emergence altogether. However, this strategy is unsustainable due to its reliance on fossil fuels, heavy machinery, and a large volume of environmentally harmful chemicals that are also limited in quantity on Earth. These tools also come with a significant economic cost. Moreover, there are growing concerns about the health and well-being of farmers and rural communities, as exposure to these chemicals can lead to diseases like cancer and developmental issues in children [6].

IoT can be a valuable tool for detecting and preventing the development of pests and diseases, and many projects have been developed in this domain. Comprehensive review papers on the topic have been authored by [112, 113, 114]. There are different methods to detect diseases or pests, which can be direct or indirect. Indirect methods predict the risk of disease or pest development based on environmental parameters. For instance, models to detect conditions conducive to disease development in grapes based on temperature, humidity, and leaf wetness levels have been proposed by [115] and [116]. Similar approaches have been developed for strawberries [117], general models for leaves [118], cereal crops [119], and orchard monitoring [120]. These methods often employ AI, machine learning, and knowledge engineering to identify patterns in environmental changes that lead to pest or weed development [121, 122]. Once a risk is identified, farmers are alerted but may

still need to visit the field to confirm the presence of pests or diseases. To mitigate this, drones, especially Unmanned Aerial Vehicles (UAVs), can be utilized [123]. Drones can independently survey fields, capturing images of crops to process with AI computer vision algorithms. Drones can either routinely monitor all crops [124] or focus on areas deemed at risk by indirect methods [125]. UAVs can also be programmed to apply localized countermeasures, such as spraying chemicals on affected crops and their immediate surroundings [126]. Precision application methods have proven highly effective, allowing some farms to reduce chemical usage by up to 90% [127]. Finally, direct measurements can also be taken using static sensors, such as cameras placed around fields to capture images of crops and detect diseases, like in tomato fields [128], or microphone sensors designed to count insects [129].

4. *Crops Monitoring, Yield Forecasting and Harvesting time estimation*: Local sensors and remote observation enable farmers to monitor the growth process of their crops. By doing so, they can estimate future yield and determine the optimal harvesting time. Comprehensive studies on such applications have been conducted by the authors in [130, 131, 132].

In crop monitoring, the objective is to use available data to estimate the growth stage and health status of the plants. This not only allows farmers to predict yield and decide on the best harvesting time but also helps in managing resources like fertilizers and water. For instance, the authors in [133] introduced an IoT platform utilizing a camera system and the machine learning computer vision algorithm YOLO [134] to oversee apple growth. If apples are deemed undersized for their growth period, the platform recommends fertilizer application. Once the apples are ready for harvest, an estimate of the available quantity is relayed to the farmer. The primary technology for sensing growth stages is camera-based. Cameras can be static sensor nodes or mounted on drones, typically UAVs [124]. However, for observing extensive areas like wheat fields, remote sensing is often preferred. This is typically conducted by planes or satellites. With advanced imaging techniques, they can gauge the quantity and characteristics of plants in a field, such as type, growth stage, health, and harvesting time. Commonly, they utilize the NDVI index [135]. The Normalized Difference Vegetation Index is a straightforward indicator used to determine if the observed area contains live green vegetation. Several other indices can be applied to crops, and an in-depth review on remote sensing technologies was presented by the authors in [136].

Regarding crop yield prediction, IoT platforms merge local and remote data to estimate the approximate quantity of produce available at har-

vest. For example, the authors in [137] developed a system based on NDVI and meteorological data to predict wheat production in Europe, showing promising outcomes. Yield forecasts enable farmers to make informed decisions about the types of crops they should cultivate in specific environments. They can also assess the economic worth of a land parcel. Moreover, such forecasts assist agro-industry stakeholders in predicting required logistics and potential results [138]. A detailed overview of methods for yield forecasting was provided by the authors in [139], they mainly uses Machine learning algorithms to determine insightful patterns. Crops monitoring is the one of the application that is the most implemented industry-wise. Microsoft even propose its own platform FarmBeats for monitoring [140] just as Amazon and its AWS platform.

5. *Indoor farming and controlled environment agriculture*

Indoor farming and controlled environment agriculture (CEA) refer to the cultivation of plants and crops in locations where environmental conditions—such as temperature, humidity, light, and CO₂ concentration—are closely monitored and regulated, often independent of external weather conditions [141]. This stands in stark contrast to traditional agriculture, which is heavily dependent on external climate conditions and seasonal changes [142]. Indoor farming is often carried out in greenhouses, where sunlight is the primary light source. One of the principal advantages of indoor farming is its capacity for year-round production. Traditional outdoor farming is frequently at the mercy of unpredictable weather patterns and seasonal changes. By transitioning agriculture indoors, farmers can bypass these challenges, ensuring a consistent crop yield regardless of the time of year. Additionally, because indoor farming systems can be established virtually anywhere, including in urban settings[143], they can significantly reduce the need for transporting food over long distances, leading to fresher produce and decreased carbon emissions. For example, the city of Singapore is investing a considerable amount of money into this method [144]. Moreover, controlled environment agriculture could pave the way for opportunities in space travel and colonization [145].

Indoor farming uses substantially less water than its outdoor counterpart, especially when integrated with systems like hydroponics. Pesticides and herbicides, which are often concerns in traditional agriculture due to their environmental impact, can be minimized or even eliminated in a controlled environment because of the reduced risk of pests and diseases. Controlled environment agriculture is optimally managed by IoT devices, making it one of the most fitting use cases for this tech-

nology. Temperature sensors, paired with air conditioning units, can ensure the correct temperature is maintained [146]. Ultra-Violet sensors can gauge the level of available light and control LED panels to provide the right amount of luminosity for the plants [147]. Comprehensive surveys on IoT projects concerning indoor farming have been proposed by various authors in [148, 149, 150].

However, the main limitations of indoor farming include its demand for vast amounts of energy and the need for complex infrastructure, which can make large-scale implementation challenging [151]. Additionally, not all crops are suited for indoor growth. This is especially true for grain agriculture [141].

6. *Livestock management:*

Livestock management is also improving through smart farming. Although animal product consumption should be reduced, as mentioned in the introduction chapter of this thesis, IoT offers various tools to assist farmers in managing cattle. Comprehensive surveys on IoT applications for livestock have been proposed by authors in [152, 153, 154]. The primary application is health monitoring of animals [155]. Using smart wearable tag sensors placed on cows, IoT platforms can monitor vital signs, such as body temperature or heartbeat, and alert the farmer if any anomalies, like fever, are detected [156]. One of the earliest use-cases of IoT systems for smart farming applications was the use of GPS tracking collars for livestock. This allowed farmers to track the movement of their cattle herd and create virtual fences [157]. The collar detects the GPS position of the animal and if it is in a prohibited area, sends a mild electric shock to guide the animal back to allowed zones. This approach not only allows farmers to manage their cattle without constructing extensive fence infrastructure but also enables them to dynamically modify the areas where they want their cattle to graze, potentially restricting certain zones to allow for grass regeneration. GPS collars can also be used to detect behaviors indicative of diseases, such as mastitis [158] or even detect cattle theft in certain cases [159]. Another interesting application is the detection of estrus in female animals using machine learning [160]. A valuable application is animal identification through facial recognition as proposed in [161].

Overall, IoT can assist farmers in enhancing the well-being of their livestock, thereby producing higher-quality animal products. Regrettably, this facet of smart farming also encompasses darker applications such as Animal Factory management. An industrial livestock factory, often termed a factory farm or Concentrated Animal Feeding Operation (CAFO), is a massive industrial agriculture facility where numerous livestock are

primarily raised for food production. These operations prioritize production output and efficiency, frequently at the expense of conventional farming practices or rigorous animal welfare considerations. Similarly to Controlled Environment Agriculture, IoT can be beneficial in such domains. For instance, to mitigate animal stress resulting from lifelong confinement in cramped indoor spaces, often devoid of natural light, authors in [162] have suggested virtual reality applications for cows and pigs to alleviate stress.

A noteworthy application of IoT related to animals is the monitoring and defense against wildlife. Outdoor fields frequently face threats from wild animals such as wild boars, deer, and foxes. IoT projects using computer vision have been developed to deter wild pigs when detected in the vicinity, employing alarms, as discussed by authors in [163]. Moreover, IoT can be utilized to monitor endangered species like birds, which can also serve as significant environmental indicators for agriculture, as elaborated upon in [164].

7. *Transport, storage and distribution*: The logistics of agricultural products after harvest fall outside the scope of this thesis, which exclusively addresses the farming processes within the agriculture industry. However, given that over a third of food produced is wasted [165], enhancing the logistics of food distribution remains of paramount importance. IoT can also be employed to monitor storage, such as wheat silos [166], ensure the traceability of products through RFID [167], or even automate the purchasing process by restocking fridges when they are empty or lack certain types of food [168]. Comprehensive surveys on supply chain management of food products facilitated by IoT are provided by the authors in [169, 170].

8. *Agroecology*:

Agroecology integrates the science of ecology with agriculture to create resilient, sustainable farming systems. It emphasizes the relationship between plants, animals, humans, and the environment within agricultural landscapes, aiming to enhance biodiversity, improve soil health, and promote a balance between food production and ecosystem preservation. In other words, the principle of agroecology emphasizes the interactions among various stakeholders of a farm to ensure its long-term sustainability. This entails avoiding dependence on unsustainable methods such as the use of chemical inputs, heavily mechanized oil-powered machinery, and standardized crop strains commonly associated with monocropping techniques as mentioned in 1.1. However, adopting agroecological practices presents certain challenges. These include heightened sensitivity to environmental factors (for instance,

crops may become more susceptible to drought if irrigation systems are avoided to prevent groundwater depletion and the resultant ecological concerns) and also a significant requirement for mechanical labor (labor that is usually performed by human as oil-powered machine are not only unsustainable but also not effective for fields using monocropping for example) [66]. Despite the two aforementioned drawbacks, agroecology, in certain scenarios, can be not only sustainable but also more productive per unit of land [171]. Fundamentally, agroecology mirrors pre-industrial revolution agricultural practices, wherein farmers primarily relied on their local environment for food production [172]. This paradigm has spurred significant research into smart farming, with the goal of producing efficient tools that facilitate the implementation of agroecology [33]. The underlying rationale is straightforward: rather than employing smart farming tools to render unsustainable practices marginally more sustainable, it is preferable to use these tools to enhance the productivity of inherently sustainable agriculture. Several comprehensive surveys on smart farming tools tailored for agroecology have been conducted [173, 64, 174]. Addressing the environmental sensitivity of agroecology, IoT platforms are ideally positioned, given their capacity to effectively interpret and forecast climatic scenarios using data harvested both locally and remotely. For instance, the authors in [175] introduced AI algorithms designed to decipher intricate data streams from diverse agricultural sensors, leading to improved local farm climate predictions. These predictions can subsequently guide farming strategies and crop scheduling. Taking a more pragmatic stance, the authors in [176] developed and trialed an IoT platform named SysAgria on a farm. This system offers exhaustive, real-time environmental updates and development conditions across various phenological phases of crops, fruit trees, vines, and vegetables. These insights can shape sustainable proactive treatments, fertilization planning, and the scheduling of sowing and harvesting activities. Agroecology can also be realized within greenhouses to prolong the production season. Here again, IoT proves instrumental. As demonstrated by the work in [177], IoT can meticulously regulate greenhouse irrigation (sourced exclusively from rainwater collection on the greenhouse itself) while also managing airflow, temperature, and sunlight to preempt conditions conducive to disease propagation. Regarding the mechanical labor issue, better climate prediction can help manage human resources to reduce it but the field of robotics is also a very well developed research domain within smart farming. Robotics is outside of the scope of this thesis but a good survey of robotic applications of smart farming have been conducted by the authors in [178].

9. *Small-Scale*: As highlighted in 1.1, small-scale farms (SSFs), typically encompassing an area of less than 2 ha and characterized by limited incomes, constitute 80% of global farms, as illustrated in Figure 1.9. Despite this, much of the research in Smart Farming does not take these factors into account, with a majority of projects targeting larger agricultural operations [57]. Notably, small-scale farms are often considered more sustainable than their larger counterparts [58]. Analogous to agroecology, SSFs depend heavily on manual labor and are acutely vulnerable to climate change. In response, several IoT projects tailored to the needs of SSFs have been initiated. Comprehensive surveys on this subject have been conducted by the authors in [179, 180, 181]. Predominantly, we observed that projects designed for SSFs mirror those proposed for larger farms but with an emphasized commitment to technological accessibility. This entails cost-effective development, open-source publication, and user-friendly solutions catering to farmers less adept with technology. For instance, researchers at Dedan Kimathi University of Technology introduced a budget-friendly IoT platform to automate greenhouse operations for vegetable cultivation in Kenya [182]. On a broader scale, certain projects aim to shield farmers from climatic adversities such as floods. In Tanzania, the rainy season can trigger rapid river level rises and devastating floods. The devised IoT solution comprises solar-battery powered SONAR level sensors stationed at various river locations, linked via LoRa to several grid-connected gateways with cellular connectivity. If the sensors detect swift river level increments, an SMS-alert mechanism activates, prompting downstream farmers to extract river water to accommodate the impending upstream surge. A pivotal factor underpinning the success of this system is the fervent commitment of volunteers who coordinate activities, sustain the IoT framework, and facilitate workshops to engage local communities. This not only minimizes costs but also augments awareness regarding Smart Farming applications for SSFs [183].

2.2.3 . Limitations

Despite its huge potential, IoT technology faces certain limitations regarding its technical implementation and its application to agriculture. Authors in [35] described the main issues in depth in their survey for more details. The main limitations of IoT are the following:

- **Network and communication:** IoT devices require a reliable communication medium to transmit and receive data. While wired connections offer the highest reliability in data transmission, the necessary infrastructure is often incompatible with many applications. Consequently, most IoT devices utilize wireless networks. However, wireless commu-

nication presents challenges. Firstly, devices must be within an area with network coverage, necessitating a specific gateway infrastructure. Secondly, IoT devices are typically deployed in large numbers within confined areas. This high density can lead to Quality of Service (QoS) issues, which undermine the reliability of a platform, especially when latency is a critical criterion. Various protocols at various layers of the communication stack are tailored to different applications, and a concise survey of these has been presented by the authors in [184].

- **Processing power:** The logical component responsible for processing on the devices is typically a microcontroller with limited computational power. While this reduces both cost and energy consumption, such processing units are not capable of implementing complex algorithms. As a result, there is a heightened reliance on the cloud computing capabilities of IoT platforms, which in turn increases communication requirements, as highlighted by the authors in [185].
- **Energy consumption:** Devices are typically battery-powered since a wired connection can lead to excessive infrastructure costs. This necessitates the devices to be optimized to run for multiple years without the need for battery replacement. As a result, energy consumption becomes a pivotal criterion for IoT devices. Factors that influence energy consumption include computation time, type of communication, and its duration. Wireless transmission is often the primary energy-consuming activity, depending on the protocol used [186]. It's also worth noting the emergence of organically powered batteries like BeFC, which utilize glucose to generate electricity [187]. These batteries are fully biodegradable and sustainable, but they offer only limited energy, making them suitable only for energy-efficient systems. Conversely, efforts have been made to equip IoT devices with energy harvesting components that tap into the energy available in their environment, such as vibrations [188].
- **Data Storage:** IoT platforms produce vast amounts of data. This necessitates dedicated storage infrastructure and efficient management [189].
- **Data interpretation:** Depending on the application domain, interpreting vast amounts of diverse data can be challenging. The decision-making component of an IoT platform must be efficient enough to deliver impactful results for its intended domain. While decision algorithms can be as basic as conditional statements like "if/else," it is more beneficial for the platform to incorporate domain-specific knowledge to make more insightful decisions [190].

- **Security and Privacy:** Many IoT devices lack adequate security protocols, often due to limited computational power, rendering them vulnerable to breaches, particularly to "man-in-the-middle" attacks during firmware updates. As the number of devices multiplies, the potential attack surface expands exponentially. Beyond security issues, privacy remains a significant concern. Continuous data collection by devices raises alarms about surveillance, personal privacy, and the broader societal implications of such an interconnected world. Consequently, safeguarding this data's privacy is crucial. If the data isn't encrypted or securely stored, it becomes an enticing target for hackers [191].
- **Interoperability:** Many IoT devices are manufactured by different companies using various standards. This can lead to devices being incompatible with one another. Same issue with communication protocols as their multiplicity can be difficult to manage. Without a unified standard, integrating different devices to work seamlessly in an ecosystem can be challenging [192].
- **Cost:** The cost for setting up the necessary infrastructure for IoT, such as sensors, networks, storage, and processing units, can be expensive. It is important that IoT systems offers a economically viable implementation for companies. In addition to the setup expenses, there are maintenance costs to consider. Over time, devices might require updates, repairs, or replacements, all of which contribute to the cumulative expenses [193]. Furthermore, the environmental cost should not be overlooked. As the number of devices grows, so does electronic waste. Proper disposal becomes a concern [194].
- **Regulation:** Different countries have different regulations concerning data privacy, device standards, wireless communication bandwidth etc. Ensuring compliance across regions can be complex. In addition there is liability issues. If an IoT device fails and causes harm or damage, determining liability can be complex [195].

In the context of agriculture, IoT faces unique challenges as exposed by the authors in [196, 197].

On the hardware side, the often remote and open nature of farming environments can result in connectivity issues, particularly in vast rural fields where network access may be inconsistent. Devices might be situated in locations that are difficult to access, especially during the growing season (consider a sensor placed in the midst of a cornfield). Consequently, it's vital to ensure their reliable operation over extended periods, which predominantly hinges on efficient energy management. Furthermore, basic environmental

sensors (like those for temperature, wind speed, and rainfall) may not provide sufficiently insightful data. Computer vision algorithms, while effective for various applications, demand increased computational power, which augments the reliance on cloud computing and communication — this, in turn, can drain batteries more rapidly. Lastly, agricultural IoT devices are subject to the rigors of the elements, necessitating durable, weather-resistant designs.

On the software side, decision-making algorithms must process and interpret a broad spectrum of environmental data. Making sense of this data with rudimentary algorithms can be challenging. For instance, the irrigation system described in part 1.1 and illustrated in figure 1.6 operates the water valve based solely on readings from a soil moisture sensor. This approach is suboptimal since irrigation procedures should also account for current and forecasted weather, soil type, crop variety, calendar, sun exposure, sensor type, manufacturer, firmware version, and more. This diverse range of factors complicates the decision-making process.

Additionally, the upfront costs and specialized expertise needed to deploy IoT systems can be prohibitive for farmers with limited resources. Tackling these agriculture-specific challenges is crucial to fully harness the potential of IoT in smart farming.

Moreover, the initial investment and technical know-how required for implementing IoT systems can be a barrier for resource-constrained farmers. Addressing these agricultural-specific limitations is essential to fully unlock the benefits of IoT in Smart farming.

2.2.4 . Amelioration axes

All the mentioned limitations have a range of proposed solutions that are described in the associated cited papers. In this thesis, the focus is directed towards optimizing the energy efficiency of sensors and deploying a knowledge-based decision system. This system is designed to make the sensors last as long as possible and to take the best judgments, considering both the data from sensors and the principles of agroecology farm management. Initially, energy efficiency is tackled in a conventional manner by adopting dedicated wireless communication protocols. The objective then shifts to bypassing the cloud for executing complex algorithms, such as computer vision, in order to reduce the substantial energy costs associated with data transfer. To achieve this, a new AI domain termed "Embedded AI" [198] seeks to bring computational processes directly to the end devices, thus eliminating data transfer and enhancing privacy concurrently. Finally the aim is to employ knowledge engineering to assist the system in making the most informed decisions, especially regarding Agroecology which is a complex field with a vast amount of parameters to take in account. Consequently, these three areas—energy-efficient wireless sensor communication, Embedded AI, and knowledge engi-

neering—are the focal points of the subsequent sections in this state-of-the-art review.

2.3 . Energy efficient Wireless Sensor Network

2.3.1 . Definition

A Wireless Sensor Network (WSN) is a collection of spatially distributed sensors that cooperate to monitor physical or environmental conditions, such as temperature, sound, images, pressure, humidity, etc. These sensors can autonomously communicate the information through wireless protocols to a main location (typically a gateway or base station) which then processes the data towards an application server, either locally or in the cloud, for the data to be exploited. Even though we focus on the sensors in WSNs, from the IoT point of view, actuators are also to take in account as wireless nodes on the networks as their communications will also weight on the network behavior. Good reviews and architectural descriptions have been proposed by the authors in: [199, 200, 201]. A typical layout of multiple wireless sensor network is depicted in figure 2.6 where we can see the different components:

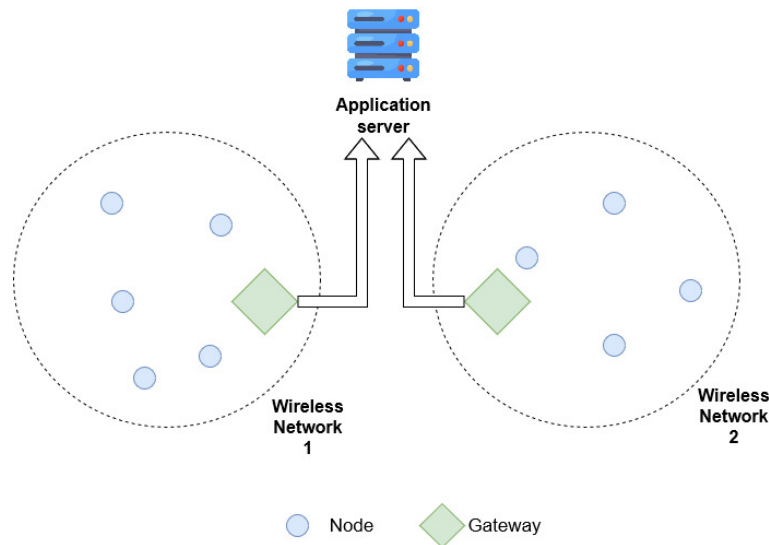


Figure 2.6: Typical Wireless sensor networks architecture

- **Nodes:** These are the main components of the network. Each node is typically equipped with a radio transceiver, a microcontroller, an interfacing circuit, and an energy source, usually a battery. In addition there is a sensory part that captures the data from the environment (such as camera, temperature probe, etc.) and/or an actuators such as a light or a motor.

- **Base Station/Gateway:** This is the component that gathers data from the sensor nodes, processes it, and might forward the data to other networks (like the internet). The base station is typically more powerful in terms of computational capabilities and energy than the individual sensor nodes. It can also be called sink.
- **Application server:** The Application server (or servers) is the end point of the data where they are processed and analysed to produce interesting output. It can be located on the local network of the WSN or in the cloud.

A WSN has also various core characteristics :

- **Wireless Communication type:** Nodes communicate with each other or with a base station through wireless protocols. There is a variety of protocols available with different pros and cons regarding the network characteristics. One or more wireless protocols can be used depending on the nodes, thus creating heterogeneous WSN. We will be describing those protocols more in details in part ??.
- **Quality of Service (QoS) [202]:** It refers to the set of techniques and mechanisms used to manage network resources, ensure the performance, reliability, and priority of specific data flows, and meet the requirements of different types of network traffic. It also measures key network performance metrics. The main components of QoS are:
 - *Bandwidth:* Refers to the maximum rate of data transfer across a network path. QoS can allocate specific bandwidth amounts to different types of traffic to ensure smooth flow.
 - *Latency:* The time it takes for a packet to travel from source to destination.
 - *Packet Loss percentage:* Refers to the number of packets that are sent but never arrive at their destination. This can happen due to network congestion, faulty hardware, or other issues.
 - *Signal to Noise Ratio (SNR):* In communication, SNR is a measure used to describe the level of a desired signal to the level of background noise. It represents how much the intended signal stands out from unwanted background noise. A higher SNR indicates a clearer and better-quality signal, while a lower SNR suggests that the signal is harder to distinguish from the noise.
 - *Jitter:* Variation in the delay of received packets. Real-time voice and video communication can be seriously affected by high jitter values, as it results in choppy or disrupted streams but this is less important for nodes in a WSN.

- *Congestion and traffic management:* These mechanisms control the rate of traffic flow. Policing drops or marks packets that exceed a specified rate, while shaping buffers and delays excess packets to fit within a defined rate. During peak times, a network might experience congestion. QoS mechanisms help in managing this congestion, ensuring that priority traffic gets through while lower-priority traffic might be queued or even dropped.
- **Topology [203]:** Depending on the application and network design, WSNs can take on different topologies. The most common ones for WSNs are star, tree, or mesh topologies and are depicted in figure 2.7.

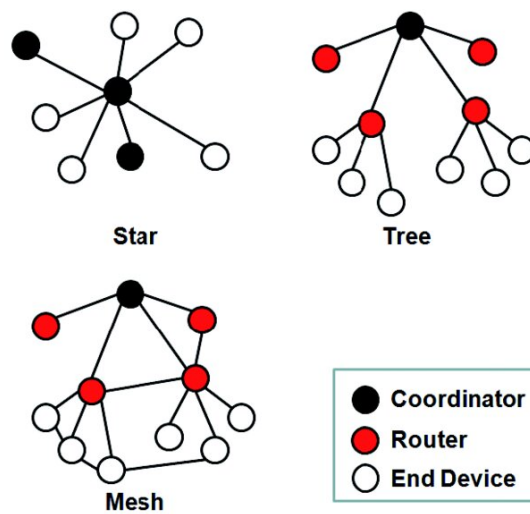


Figure 2.7: WSNs topologies [204]

- **Star topology:** In star topology, the coordinator or hub is a single central node. In the network every node is connected to the hub. Star topology is very easy to implement, design and expand. As all the data flows through the hub, it plays an important role in the network and a failure in the hub can result in failure of entire network. Single-hop network are star topology where the hub act as the gateway.
- **Tree topology:** A tree topology is a hierarchical structure that organizes nodes in a parent-child relationship, resembling the structure of a tree. At the top, there's typically a single node, often referred to as the coordinator, root or base station. Each node, except the root, has one (and only one) parent node and zero or more child nodes and the nodes that have child nodes are referred to as interior nodes, while nodes without any children are called

leaf nodes or leaves. The processing power and energy consumption is highest at the root node and keeps on decreasing as we go down the hierarchical order. Tree topologies are especially suitable for applications where data is collected in a hierarchical manner but are susceptible to failure as if an interior node fails, all of its child nodes can lose connectivity to the root.

- **Mesh topology:** In a mesh topology, every node not only transmits its own data but also functions as a relay for data from other connected nodes. There are two types of mesh topologies: Fully Connected Mesh and Partially Connected Mesh. In the fully connected mesh topology, every node is connected to every other node. Conversely, in the partially connected mesh topology, a node is connected only to one or more neighboring nodes. Mesh networks offer several advantages. First, they allow for network coverage expansion without the need for costly gateways, relying instead on nodes that can also function as relays. Secondly, if a relay node or gateway fails, the network can reconfigure its routes to continue transmitting messages. The issue is the increased energy consumption of certain nodes that will have to perform networking task even when not in use for their primary role (sensing or actuating).
- **Scalability [205]:** WSNs can range from a few to thousands of nodes. It's crucial for the network's design to handle this variability in size to insure the quality of service. Not every wireless protocols is fitted for every types of nodes density.
- **Security [206]:** WSNs are vulnerable to various security threats due to their inherent characteristics such as wireless communication, limited resources, and distributed deployment. Such threats must be addressed to insure the good behavior of the network.
- **Addressing [207]:** Addressing in WSN is a critical issue due to the limited resources of the nodes and the dynamic nature of the network. Each node in the network requires a unique identifier, called an address, to enable communication with other nodes in the network. One of the main challenges in addressing WSN is the limited address space. Due to the large number of nodes and the limited memory and processing power of the nodes, addressing schemes in WSN need to be designed to use as few bits as possible while still providing a unique address to each node. Another challenge is the dynamic nature of the network. Nodes in the network may move or fail, causing changes in the network topology. This can lead to problems with address assignment

and resolution, which can result in communication failures or increased latency.

- **Firmware update [208]:** Firmware updates WSNs can be challenging due to the limited resources of the sensor nodes and the distributed nature of the network. Firmware updates are necessary to improve the functionality and security of the network and to address bugs and vulnerabilities. However, updating firmware in WSNs requires careful consideration of several issues. One of the main challenges is the limited storage and processing power of the sensor nodes. Firmware updates typically require significant amounts of memory and processing power, which can be a problem for resource-constrained nodes. Updating firmware on all nodes in the network can also consume a significant amount of network bandwidth and energy, leading to network congestion and reduced network lifetime. Another challenge is the distributed nature of WSNs. Sensor nodes are typically deployed in remote and hard-to-reach locations, making it difficult to update firmware on all nodes simultaneously. This can result in inconsistency in the firmware versions across the network, which can lead to compatibility issues and communication problems. To address these challenges, various approaches have been proposed for firmware updates in WSNs. These include incremental updates, where only the parts of the firmware that have changed are updated, and delta encoding, where only the differences between the current and new firmware versions are transmitted. These approaches can reduce the amount of memory and bandwidth required for firmware updates. Another approach is to use over-the-air programming (OTA), which allows firmware updates to be transmitted wirelessly to the sensor nodes. OTA updates can be more efficient than physically updating each node and can be done remotely, reducing the need for on-site maintenance.
- **Cost [49]:** Cost is a significant issue in WSNs due to the large number of nodes that are typically deployed and the limited resources of the nodes. One of the main cost factors in WSNs is the cost of the sensor nodes themselves. Sensor nodes can vary in cost depending on the type of sensor used, the processing power, and the communication capabilities. Deploying a large number of nodes can quickly become expensive, making it challenging to deploy WSNs on a large scale. Another cost factor is the cost of maintaining the network. WSNs typically require ongoing maintenance, including battery replacements, firmware updates, and physical repairs. Maintaining a large number of nodes can be challenging and costly, especially in remote or hard-to-reach locations. Additionally, the cost of data storage and processing can be significant in

WSNs. Sensor nodes generate a large amount of data, and processing this data in real-time can be computationally intensive. Storing and processing this data requires specialized hardware and software, which can be expensive.

Finally, we emphasize the energy consumption of such networks. WSNs must be energy-efficient as devices run on batteries and need to last extended periods of time. An energy-efficient WSN prioritizes the battery lifespan of the device, often at the expense of data rate, range, quality of service, etc. A good survey on energy management methods in WSNs have been proposed by the authors in [209].

2.3.2 . WSN communication protocols

Wireless communication refers to the transfer of information or data between devices or systems without the use of physical connections, such as wires or cables. Instead, wireless communication utilizes electromagnetic waves, which include radio frequencies, infrared signals, microwaves, and other forms of electromagnetic radiation. In WSNs, when we talk about communication protocols we refer to the overall communication stacks as described in the OSI and TCP/IP model and not only the wireless physical transmission of data. The OSI (Open Systems Interconnection) model is a conceptual framework used to understand how different networking protocols interact across diverse networking infrastructures [210]. Developed by the International Organization for Standardization (ISO) in the late 1970s, the model defines a set of seven layers, where each layer represents a specific set of functions necessary for communication systems. These layers range from the physical hardware components used to transmit bits over a medium to high-level data formatting and presentation. The key idea behind the OSI model is that different networking tasks are compartmentalized into specific layers, allowing for modularity and making it easier to understand and troubleshoot network operations. The TCP/IP model, sometimes called the Internet protocol suite, is the foundation for modern internet communication [211]. Developed also in the 1970s by the U.S. Department of Defense, it has become the dominant framework for networked communication. Instead of the seven layers in the OSI model, the TCP/IP model consists of only four layers. These layers are more broadly defined than the OSI layers, making the model less granular but more aligned with actual protocol design and implementation. While it's not entirely accurate to say the TCP/IP model was "derived" from the OSI model, there's a general understanding that the design of the OSI model influenced the articulation and documentation of the TCP/IP model. The latter was developed before the OSI was standardized but was formally documented afterward. The TCP/IP model can be mapped to the OSI model's layers, even though they don't align perfectly. This mapping is often used in teaching and discussing

network protocols to show the equivalences and differences between the two model. Those alignments are described in figure 2.8. The Application Layer in TCP/IP encompasses the functions of the Application, Presentation, and Session layers of the OSI model. The Transport Layer in TCP/IP corresponds directly to the Transport Layer in the OSI model. The Internet Layer in TCP/IP aligns with the Network Layer in the OSI model. Finally, the Network Access Layer in TCP/IP combines the functions of the Data Link and Physical layers of the OSI model.

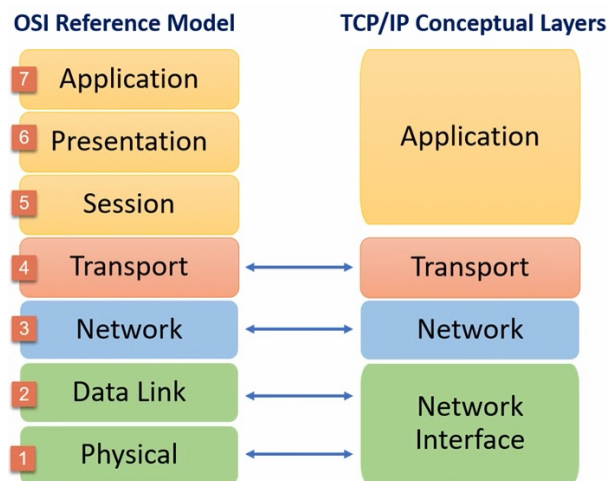


Figure 2.8: OSI and TCP/IP

In the following section, we will describe the main protocols used in the domain of IoT for the different layers of the TCP/IP model. As mentioned previously, there is no standard for all IoT applications; thus, there is a vast array of protocols available that address various issues. Our goal here is not to provide a comprehensive survey for each layer, but we will mention reliable sources for in-depth surveys for each layer and describe the primary protocols.

- **Application layer [212]:** It is the the topmost layer, and it provides the interface between the user's application and the network. It deals with end-user services and facilitates communication between software and lower layers of the TCP/IP model. The four main protocols in the IoT domain at this layer are:
 - *MQTT (Message Queuing Telemetry Transport) [213]:* It was, introduced in 1999, stands as one of the pioneering M2M communication protocols. Andy Stanford-Clark from IBM and Arlen Nipper from Arcom Control Systems Ltd (now Eurotech) were the brains behind its development. This protocol operates on a publish/subscribe

mechanism, tailored for efficient M2M communication in bandwidth-limited environments. Within this system, an MQTT client sends messages to an MQTT broker. Other clients can then subscribe to these messages or even store them for future access. Messages are directed to specific addresses called topics, and a client can subscribe to multiple topics, receiving all messages associated with each one. MQTT operates as a binary protocol, typically featuring a 2-byte fixed header, and can handle message payloads up to a maximum of 256 MB. It employs TCP for transport and fortifies security using TLS/SSL. Hence, the communication between the client and the broker remains connection-based. A standout feature of MQTT is its three-tiered Quality of Service (QoS) ensuring message delivery reliability. It's ideal for extensive networks with numerous small devices requiring back-end internet server monitoring or control. However, MQTT isn't designed for direct device-to-device transfers or broadcasting data to a multitude of receivers. Its design is fundamental, presenting only a limited set of control features.

- *CoAP (Constrained Application Protocol) [214]*: It originates from the IETF CoRE (Constrained RESTful Environments) Working Group and serves as a streamlined M2M protocol. It is designed to facilitate both request/response and resource/observe models, which is akin to a publish/subscribe system. CoAP's primary aim is to function alongside HTTP and the RESTful Web, enabled by straightforward proxies. In contrast to MQTT's topic-based approach, CoAP employs Universal Resource Identifiers (URIs). In this system, publishers dispatch data to URIs, while subscribers align themselves with specific resources defined by these URIs. When publishers push new data to a URI, every subscribed party receives a notification regarding the updated URI value. CoAP operates as a binary protocol, typically having a 4-byte fixed header. Its message payload sizes are contingent on the web server or the underlying programming framework. CoAP's foundation is built upon the UDP transport protocol and it adopts DTLS for its security measures. As a result, communication between clients and servers occurs through connectionless datagrams, which might compromise reliability. However, to address this, CoAP introduces two distinct QoS levels via "confirmable" and "non-confirmable" messages. Receivers must acknowledge "confirmable" messages with an ACK packet, whereas "non-confirmable" messages don't demand any acknowledgment. CoAP provides a richer feature set compared to MQTT. For instance, it incorporates content negoti-

ation, enabling clients to specify how they'd prefer a resource to be represented. This flexibility ensures that both client and server can evolve separately, allowing for the introduction of new representations without causing mutual disruptions.

- *AMQP (Advanced Message Queuing Protocol) [215]*: AMQP is a lightweight M2M protocol formulated by John O'Hara at JPMorgan Chase in London, UK, back in 2003. Designed with a corporate mindset, its architecture prioritizes elements like reliability, security, provisioning, and interoperability. AMQP can cater to both request/response and publish/subscribe models, offering a plethora of messaging-associated features like reliable queuing, topic-based messaging, flexible routing, and transactions. Central to AMQP's communication mechanism is the concept of an "exchange." Either the publisher or the consumer has to establish this "exchange" by naming it and then circulating this name for discovery purposes. Following this, the consumer crafts a "queue" and immediately links it to the created exchange. For a message at the exchange to reach its intended destination, it undergoes a "binding" process to match with the right queue. There's versatility in how AMQP can distribute messages – directly, via fanout, based on topics, or leaning on headers. AMQP's structure is rooted in a binary protocol, typically showcasing an 8-byte fixed header. The message payload sizes, though, vary depending on the broker/server or the development framework in place. By default, AMQP relies on TCP for transportation, supplementing its security using TLS/SSL and SASL. Communication in this system is connection-driven, between the client and the broker. One of AMQP's cornerstone features is its reliability. To this end, it provides two foundational QoS levels for message delivery: the Unsettle Format, which isn't inherently reliable, and the Settle Format, which ensures reliability.
- *HTTP (HyperText Transfer Protocol) [216]*: HTTP, primarily known as a web messaging protocol, was an original creation of Tim Berners-Lee. It later underwent joint development by the IETF and W3C, culminating in its official standardization in 1997. The core of HTTP lies in supporting the request/response architecture of the RESTful Web. Drawing a parallel with CoAP, HTTP too utilizes Universal Resource Identifiers (URIs) rather than topics. Within this structure, servers dispatch data via URIs while clients retrieve data through specific URIs. HTTP's foundation is a text-based protocol. Notably, it doesn't set definitive specifications for the size of headers or message payloads; these parameters are influenced by the associated web server or development technology. As for trans-

port, HTTP predominantly leans on TCP, enhancing its security using the TLS/SSL protocol. This ensures that interactions between the client and server are consistently connection-driven. Although HTTP doesn't intrinsically lay out QoS parameters, incorporating them demands supplementary solutions. Recognized worldwide as a pivotal web messaging standard, HTTP introduces a range of features, encompassing persistent connections, request pipelining, and chunked transfer encoding.

The choice between one protocol and another is typically determined by various factors such as message size, resource requirements, power consumption, latency, bandwidth availability, and reliability. Authors in [217] have proposed a method to decide based on these criteria.

- **Transport Layer [218]:** The transport layer in the TCP/IP model serves as the mediator for end-to-end communication between devices on a network. It manages the transmission of data, ensuring that the information sent from a source device is accurately received by the destination device. By doing so, the transport layer facilitates seamless communication and data integrity across the network, irrespective of the underlying physical infrastructure or the geography between the communicating devices. Two of the most well-known transport layer protocols are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).
 - *TCP (Transmission Control Protocol):* TCP is a connection-oriented protocol, which means that it establishes a dedicated communication path between two devices before data transfer begins. This protocol ensures the reliable delivery of data packets by sequencing them and requiring acknowledgments for received packets. If an acknowledgment isn't received within a certain timeframe, TCP assumes the packet was lost and resends it. The primary advantage of TCP is its reliability: TCP ensures that data packets are delivered to the receiver in the correct order and without errors. Lost packets are retransmitted. However, from an IoT perspective, a drawback is that establishing a TCP connection is more complex, leading to more data transmission and subsequently increased energy consumption.
 - *UDP (User Datagram Protocol):* Unlike TCP, UDP is a connectionless protocol. It sends data without establishing a dedicated end-to-end connection. There's no guarantee that the packets sent will reach their destination or that they will do so in the correct order. However, this means that UDP has lower overhead and can deliver data faster. UDP is suitable for situations where speed is a

priority over reliability, like streaming audio or video where occasional packet loss might be acceptable. Despite UDP being faster and simpler, its unreliability makes its implementation complex in WSN scenario where nodes need to be sure that their data is transmitted.

In conclusion, TCP might be preferred for applications where data reliability and integrity are crucial, such as firmware updates, critical sensor data transmission, or device management. UDP might be more suitable for scenarios demanding real-time data delivery with minimal overhead, such as periodic telemetry from sensors, real-time monitoring, or multimedia streaming in IoT contexts.

- **Network layer [219]:** It is responsible for determining the best path to route data across a network of interconnected devices and networks. It handles the process of transmitting data packets from the source to the destination, potentially across multiple and diverse networks. The main objective of the network layer is to provide a means for data packets to traverse networks, taking into consideration factors like network congestion, topology, and outages. The core functionality of the network layer revolves around logical addressing with the help of IP (Internet Protocol). IP allows for the logical addressing of hosts and directs packets based on these addresses. IPv4 is the fourth version of IP, and for a long time, it was the dominant protocol. It uses 32-bit addresses, which limits the number of devices that can be addressed. As the successor to IPv4, IPv6 was introduced to tackle the address exhaustion issue associated with IPv4. It uses 128-bit addresses, providing a vastly larger address space. The introduction of IPv6 is an important progress towards the unique identification of each node which is fundamental for the deployment of efficient IoT architecture, but poses problems for data transmission on the physical layer due to its length. For example, the 802.15.4 wireless communication protocol has a maximum size of the PSDU (Physical layer Service Data Unit) of 127 bytes. When considering the 25 bytes from the MAC sub-layer (without security), this results in 102 bytes at the link level. Adding data link layer security, only 81 bytes remain available at the IP level. It's also important to account for the overhead from IPv6 headers (40 bytes), any potential extension headers, UDP (8 bytes), or TCP (20 bytes). In the end, the actual data payload is quite limited (33 bytes for UDP and 21 bytes for TCP), which does not meet the IPv6 specifications that require a minimum transmission unit of 1,280 bytes. This is why some WSN protocols require an adaptation layer located between the network layer and the link layer of the OSI model. For 802.15.4, this protocol is called 6LoWPAN [220]. It receives

IPv6 packets of 1,280 bytes from the network layer and transmits them to its counterpart on the remote device in 802.15.4 frames. Since these frames only have 81 bytes available, the adaptation layer must fragment the IPv6 packets before sending them and reassemble them upon receipt.

- **Network Interface/Wireless communication:** The network interface layer in the TCP/IP model, also known as the link or physical layer in other models, is responsible for the direct transmission of data between two devices on the same network. It deals with the physical connection, ensuring that data packets are placed onto the network medium and are received from it, converting the data between the packet format and the format suitable for the transmission medium, be it wired or wireless. This layer also manages error detection and correction for data being transmitted and received. The network interface layer handles the specific requirements of different communication technologies used in IoT, such as low power consumption for battery-operated devices or the ability to function in challenging environments. The wireless communication process in WSNs is the most energy costly as explained in [221]. In terms of wireless communication we make the distinction between Personal Area Network (PAN), Local Area Network (LAN) or Wide Area Network (WAN).

A PAN is a network for personal devices, typically covering a short range, usually less than 100m, like within the proximity of an individual. Famous PAN used in Wireless IoT communications are:

- *Bluetooth Low Energy (BLE) [222]:* BLE, or Bluetooth Low Energy, is a wireless communication protocol designed for short-range communication between devices while consuming significantly less power than classic Bluetooth. It also using the 2.4 GHz frequency band. Originally introduced as part of the Bluetooth 4.0 standard, BLE is optimized for situations where battery life is crucial, making it an ideal choice for many IoT devices. Due to its low power consumption, devices using BLE can operate for long periods, sometimes even years, on tiny coin-cell batteries. BLE maintains a similar communication range to classic Bluetooth, typically around 100 meters or less depending on environmental factors and hardware. While it's designed for sending smaller packets of data compared to classic Bluetooth, BLE's data transfer rates can reach up to 2 Mbps. This combination of features—low power usage, reasonable range, and efficient data transfer—has led to its widespread adoption in health and fitness wearables, smart home devices, beacons for location awareness, and many other IoT applications.

- *802.15.4* [223]: The IEEE 802.15.4 protocol is a wireless standard designed specifically for low-rate wireless personal area networks (LR-WPANs). It operates in the unlicensed industrial, scientific, and medical (ISM) frequency bands, including commonly used bands like 2.4 GHz (worldwide), 915 MHz (Americas), and 868 MHz (Europe). Depending on the frequency band, data rates can be 250 kbps (for 2.4 GHz), 40 kbps (for 915 MHz), or 20 kbps (for 868 MHz). The protocol can support various network topologies such as point-to-point, point-to-multipoint, and mesh, enhancing flexibility and range. Addressing in 802.15.4 is versatile, with both short (16-bit) and extended (64-bit) modes. Its frame structure is adaptive, catering to different types like data, acknowledgment, command, and beacon frames. Typically, the protocol can offer a range between 10 to 100 meters, though this can be influenced by the environment and specific setup. One of its standout features is its power management, optimized for battery-operated devices, allowing them to go into sleep mode to conserve energy. Security isn't overlooked either; the protocol integrates security at the MAC layer using AES-128 encryption. This blend of features positions 802.15.4 as an excellent choice for IoT scenarios requiring reliable, secure, and low-power wireless communication over short to medium distances. On this protocol are based two really famous IoT networking stacks Zigbee and Thread. Zigbee is a high-level wireless communication protocol designed for low-power, short-range applications. It stands out for its ability to support mesh networking, which enables devices to relay messages through other devices, enhancing network range and resilience. This makes Zigbee especially suitable for creating robust, self-healing networks with a large number of nodes. Zigbee's focus on low power consumption makes it ideal for battery-operated devices in Internet of Things (IoT). Thread is similar to Zigbee, but focuses on creating secure, scalable, and interoperable networks. A significant feature of Thread is its ability to form mesh networks, which enables devices to communicate with each other directly or through intermediate devices to enhance connectivity and resilience. Developed by the Thread Group, which includes notable industry players like Google, Apple, and Qualcomm, this protocol is designed with an emphasis on secure and reliable communication, especially in home automation environments. Additionally, Thread supports IPv6 natively, facilitating seamless internet connectivity and integration with cloud services.ecosystems.
- *RFID (Radio Frequency Identification)* [224]: RFID, or Radio-Frequency

Identification, is a technology that uses electromagnetic fields to identify and track tags attached to objects. These tags contain electronically stored information. Unlike barcodes, which require direct line-of-sight for scanning, RFID tags can be read through various materials from a distance. There are two primary types of RFID tags: passive tags, which have no internal power source and draw power from the RFID reader's electromagnetic waves, and active tags, which have their own power source, usually a battery, enabling them to transmit data over longer distances and even initiate communication. The data capacity, read range, and frequency band (commonly low, high, or ultra-high frequency) of RFID tags can vary. This technology is widely used in numerous applications like supply chain management, asset tracking, and access control due to its ability to operate without direct contact, thereby offering efficient and streamlined tracking and identification.

- *NFC [225]*: NFC, or Near Field Communication, is a short-range wireless communication technology that allows two devices to exchange data when they're brought close together, typically within a few centimeters. Built upon the foundations of RFID, NFC is designed for simple, secure data exchanges. It's often integrated into mobile devices, enabling functionalities like contactless payments, ticketing, and data sharing.

A LAN, is a network that connects computers and other devices in a limited geographical area, such as a home, office, or a group of buildings, typically up to a few hundreds of meters. Famous LAN used in Wireless IoT communications are:

- *802.11 [226]*: The 802.11 protocol, commonly known as Wi-Fi, is a set of wireless networking standards developed by the IEEE. It enables devices to communicate wirelessly, typically over short to medium distances. Its main characteristics include supporting multiple frequency bands (like 2.4 GHz and 5 GHz), providing various data rates up to 10 Gbps, ensuring security through encryption methods such as WPA and WPA2, and allowing seamless connectivity in home, office, and public spaces. In IoT, 802.11 is widely used for proof of concept or simple applications because such networks are usually easily available. However, Wi-Fi is not an efficient scalable network for a large number of devices. Moreover, its energy consumption is quite high, significantly affecting the battery life of nodes. Finally, its range makes it unsuitable for IoT platforms with nodes covering a vast area.

- *Z-Wave [227]*: Z-Wave is a wireless communication protocol primarily designed for home automation. It allows smart devices, such as lights, thermostats, and door locks, to communicate with each other and be controlled remotely on 900 MHz frequency bands. Z-Wave operates on low-energy radio waves with a specific range, making it efficient for short to medium-distance communications typical in home environments. Its typical data rate goes from 10 kbps up to 100 kbps. For IoT devices, Z-Wave is beneficial because it provides a reliable, low-latency communication network, can support a large number of devices in a mesh network, and consumes minimal power, essential for battery-operated devices. Furthermore, its ability to mesh means devices can relay messages for each other, extending the effective range and ensuring robust connectivity throughout a household or building. However its limited range can be an issue.

A WAN is a network that covers a broad area, such as a city, country, or even global connections. It's designed to connect multiple smaller networks, like Local Area Networks (LANs) or Metropolitan Area Networks (MANs), allowing data exchange and communication over long distances. Famous WAN used in Wireless IoT communications are:

- *Cellular Networks [228]*: Traditional cellular networks (like 3G, 4G, and the emerging 5G) are being used for IoT devices that require higher data rates and broad geographic coverage. The introduction of 5G, in particular, promises to enhance IoT with faster speeds and lower latencies. A significant limitation of cellular networks is their dependence on Internet Service Provider (ISP) companies, which are responsible for establishing network coverage through dedicated infrastructure. Additionally, they can have considerable energy demands and necessitate the use of licensed frequency bands, which implies extra costs for network access. The cost consideration becomes critical when managing an IoT fleet comprising thousands of devices. Paying for individual network plans for each device could become financially untenable.
- *Satellite Communication [229]*: Satellite communication in the realm of IoT involves using satellites to enable and ensure connectivity for IoT devices, especially in locations where traditional networks might not reach such as deep-sea vessels, forested areas, or desert regions. Despite its vast coverage, satellite communication comes with its own set of challenges. First and foremost, the cost can be substantially higher than conventional connectivity methods. Setting up the necessary infrastructure, such as satel-

lite and related equipment, can be expensive, therefore, the cost of access to such service might be high and in a similar way as cellular networks, can be prohibitive in case of large IoT Fleet. Additionally, the latency in satellite communication can be higher than terrestrial networks, meaning there can be a noticeable delay in transmitting and receiving data. This can be an issue for applications that demand real-time responses. The data rates might also be lower compared to more traditional communication methods, limiting the amount of data or the speed at which it can be transferred.

- *WiMAX (Worldwide Interoperability for Microwave Access) [230]*: It is a wireless broadband technology designed to provide high-speed internet access over long distances. Essentially, it's an alternative to cable and DSL internet services but operates over a wireless link. One of WiMAX's core technical characteristics is its ability to provide broadband connectivity over a wide area, potentially covering several kilometers, making it particularly suitable for remote or underserved regions. Operating in the microwave part of the radio spectrum, typically between 2 and 66 GHz, WiMAX can offer speeds comparable to traditional broadband services. It utilizes a technology called Orthogonal Frequency-Division Multiple Access (OFDMA) to efficiently use bandwidth and cater to multiple users. This ensures a steady and reliable connection, even when many users are accessing the network. It's also worth noting that WiMAX supports both point-to-point and point-to-multipoint access modes, which offers flexibility in how the network can be deployed and used. However, the use of WiMAX for IoT comes with certain drawbacks. Firstly, even though WiMAX can cover several kilometers, it doesn't match the widespread coverage of cellular networks, which are more prevalent. Additionally, setting up WiMAX infrastructure can be expensive, especially when considering the costs of base stations and other associated equipment. Being a line-of-sight technology, WiMAX can experience issues with interference from physical obstacles, especially in urban environments with tall buildings. This can affect signal quality and reliability. Finally WiMAX, in many parts of the world, hasn't seen the level of adoption that technologies like LTE or even 5G have. As a result, the ecosystem, in terms of devices, integration, and support, might not be as robust as those for more prevalent technologies.
- *LPWAN (Low Power Wide Area Network) [231]*: These networks are designed for long-range communications and consume minimal power, at the expense of data rate. They are particularly well-

suitable for IoT applications, and therefore will be the focus of the next section.

Figure 2.9 expose the relation between distance and data rate for each protocol.

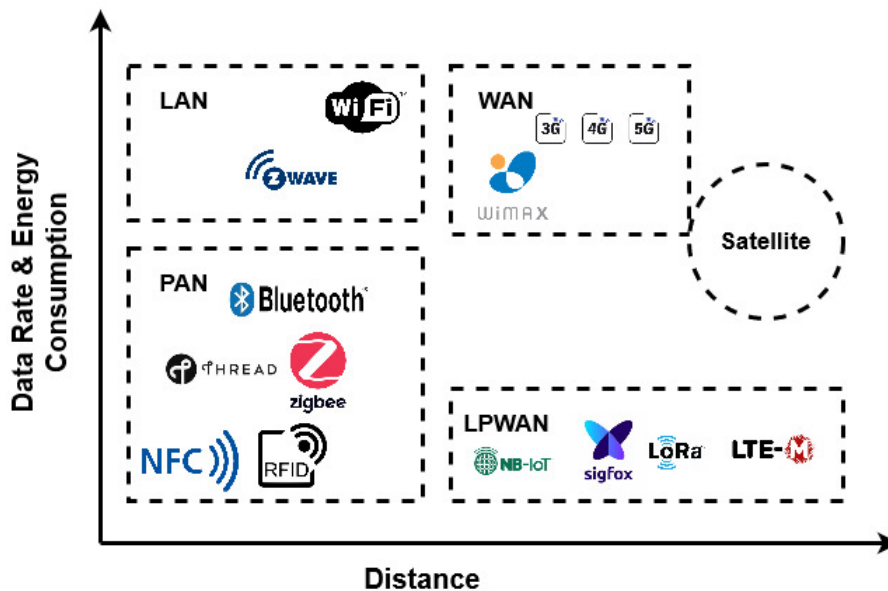


Figure 2.9: Range vs speed and energy consumption in different types of Network

To conclude this section, based on our state-of-the-art study and as showned by the authors in [232], we assert that LPWANs are the most effective wireless communication method for agricultural scenarios. Therefore, their technical specifics, applications, limitation and opportunities will be explained in greater depth in the following sections. However, we can already explain that the choice of LPWAN was made for the following reasons:

1. **Vast coverage Area:** Agricultural farms and fields are typically spread across vast areas, often in remote regions. Traditional wireless networks might not provide coverage in these areas. LPWANs, on the other hand, are designed to provide wide coverage, often spanning several kilometers. This means a single LPWAN base station can potentially cover an entire farm or even multiple farms.
2. **Low Power Consumption:** Sensors in agricultural settings might be deployed in areas where it's challenging to frequently replace batteries or provide a consistent power source. LPWANs are designed for devices with limited power sources. Devices on these

networks can operate for years on a small battery. This is ideal for agricultural sensors which can be placed in the field and left to operate for extended periods without the need for frequent maintenance.

3. **Cost-Effective:** Deploying and maintaining a wireless network over a large farm can be costly. LPWAN solutions are generally more cost-effective in terms of both infrastructure and operational costs compared to other wireless solutions, making them an attractive choice for farmers and agricultural businesses.
4. **Optimal for Low Data Rate:** Agricultural sensors, such as soil moisture sensors, temperature sensors, or weather stations, typically do not generate massive amounts of data. They send concise, periodic updates. LPWANs are optimized for such low data rate transmissions, making them ideal for this use case.
5. **Easy Scalability:** As a farm grows or if a farmer decides to deploy more sensors, LPWANs can easily scale to accommodate the increased number of devices without the need for significant infrastructure changes.

2.3.3 . Low Power Wide Area Network

Low-Power Wide-Area Network (LPWAN) is a category of wireless communication technologies specifically designed to transmit small amounts of data over long distances while consuming minimal power. This makes them especially well-suited for Internet of Things (IoT) applications where devices, often battery-powered, need to send occasional or small data packets across wide areas, potentially spanning several kilometers, without the need for frequent battery replacements. To help readers understand, consider a comparison between types of networks and real-life package delivery services: WAN can be likened to UPS, FedEx, or DHL. These services can send large packages across the world quickly, but to do so, they rely on a heavy infrastructure and need a significant amount of energy (like oil) to power their trucks and planes. LAN is analogous to your local postal service: fast and efficient within your city, but it similarly relies on infrastructure and energy. PAN can be compared to a bicycle courier within your city. They can quickly deliver medium-sized packages in a neighborhood with minimal energy costs, but their range is limited. Finally, LPWAN can be imagined as traveling pigeons. They can traverse kilometers between cities without the need for infrastructure, relying merely on seeds for sustenance. However, they can only carry really small text messages.

The ability of LPWAN technologies to transmit messages over long distances with minimal energy cost is primarily attributed to a combination of

radio frequency (RF) modulation techniques, network architecture, and protocol optimizations.

- **Radio frequency modulation:** Many LPWAN technologies use spread-spectrum techniques, which spread a narrowband signal across a wider frequency band. This helps in achieving long-range communication as the signal becomes more resistant to interference and noise. In addition, LPWAN receivers are often designed to have very high sensitivity. This means they can pick up very weak signals, allowing for longer transmission ranges. Moreover, most LPWAN technologies operate in sub-GHz bands (like 868 MHz in Europe and 915 MHz in the US), which have better propagation characteristics and can penetrate obstacles more effectively than higher frequencies. Those frequency band are usually available to use for free according to regulation as they fell under the Industrial, Scientific and Medical bands frequency bands. This allows for longer range and better indoor penetration but also cost reduction. Finally, lowering the data rate in radio transmission typically translate to longer range in RF communication because the signal can be more readily detected and decoded over noise, even at very weak signal strengths.
- **Network architecture:** Many LPWAN technologies use a star network topology, where end devices communicate directly with a central gateway. This simplifies the communication process and reduces the need for energy-intensive tasks like routing. Moreover, LPWAN nodes mostly operate in "sleep" stae, only waking up to transmit or receive data. By being active only a tiny fraction of the time, they dramatically reduce their energy consumption.
- **Protocol optimization:** LPWAN technologies are optimized for transmitting small amounts of data, which means the transmission duration is short. Shorter transmissions consume less energy. In addition, Some LPWAN solutions adjust the data rate based on the quality of the radio link. When the device is closer to the gateway, it might transmit faster, and when it's further away, it might use a lower data rate to ensure reliable communication. Moreover, LPWAN technologies often employ forward error correction techniques. This allows the receiver to correct errors in received messages without the need for retransmission, thus saving energy. Finally LPWAN protocols are often simplified, with minimal overhead. This means that most of the transmitted bits are useful data, reducing the energy required for sending protocol-related overhead.

As IoT applications rises, more and more types of LPWAN are developed. Good surveys on existing LPWAN have been proposed by the authors in [\[233\]](#),

41, 234, 231]. There is four main LPWAN technologies namely LoRa (and its associated protocol stack LoRaWAN), Sigfox, NB-IoT, LTE-M.

- **Sigfox [235]:** Sigfox operates as an LPWAN service provider, delivering comprehensive IoT connectivity through its unique patented methods. It sets up its own custom base stations with advanced cognitive software-defined radios and links them to backend servers using an IP network. Devices communicate with these stations using GFSK modulation in an extremely narrow band (100 Hz) within the sub-GHZ ISM frequency range. Sigfox utilizes open ISM frequencies. Thanks to its ultra-narrow band approach, Sigfox benefits from efficient bandwidth use and notably low interference. This results in minimized power usage, elevated receiver precision, and economical antenna configurations, at a peak data rate of just 600 bps. Sigfox only allows a limited number of messages to send per day per node to avoid network congestion. Heavy data transfer such as images is not possible. Downlink message (from the server to the node) is also limited making Sigfox unsuitable for firmware update over the air.
- **LoRa [236]:** LoRa operates on a physical layer, modulating signals in the sub-GHZ ISM band using a distinct spread spectrum method. Similar to Sigfox, LoRa employs open ISM frequencies. Its communication is bidirectional, made possible by the chirp spread spectrum (CSS) modulation, which broadens a narrow-band signal across a wider bandwidth. The signal produced exhibits low interference, granting it a high resilience against disruptions and making it challenging to detect or obstruct. LoRa employs a range of six spreading factors (from SF7 to SF12) to balance data rate and distance. A higher spreading factor yields a more extended range but at the cost of a reduced data rate, and the opposite holds true. Depending on the spreading factor and channel bandwidth, the data rate of LoRa varies from 300 bps to 50 kbps. Interestingly, LoRa base stations can simultaneously receive messages transmitted at diverse spreading factors. Each message has a maximum size limit of 243 bytes. The LoRa-Alliance standardized a communication protocol based on LoRa, named LoRaWAN, with its inaugural version released in 2015. In the LoRaWAN system, every message sent by a device is captured by all nearby base stations. This multiple reception mechanism enhances the probability of successful message retrieval in LoRaWAN. More detail on LoRa and LoRaWAN will be provided in Chapter 3 of the thesis as LoRa was the selected LPWAN for our IoT platform. Finally LoRa is capable of Firmware Over The Air Update (FUOTA) and punctual heavy file transfer.
- **NB-IoT [237] and LTE-M [238]:** Those two protocols are based on the

usage of a small portion of the available spectrum in the licensed LTE frequency bands. LTE, standing for Long-Term Evolution, is a 4G wireless communications standard developed to provide faster internet speeds compared to 3G technologies. Those two protocols are standardized by the 3GPP (3rd Generation Partnership Project) organism. The 3GPP ensures the maintenance and development of technical specifications for mobile telephony network standards. The core similarity between NB-IoT and LTE-M lies in their integration into the cellular framework. Both technologies are designed to operate within the LTE spectrum and can be deployed with minimal changes to the existing LTE infrastructure. This means that for network operators, rolling out either of these technologies often involves just a software upgrade to their existing base stations. However, the two technologies diverge significantly in their specific design and use-case optimizations. NB-IoT focuses on static or low-mobility devices and prioritizes extended coverage and deep penetration. It achieves this using a very narrow bandwidth (hence the "Narrowband" in its name), which allows it to provide connectivity in challenging environments, like deep indoors or underground. Its protocol is simplified and is particularly suited for devices that transmit small amounts of data sporadically. Due to its design, NB-IoT often achieves longer battery life and offers a cost advantage for the devices, but it sacrifices data rates and mobility support. On the other hand, LTE-M, also known as Cat-M1, offers broader application potential. While it also aims for power efficiency and extended coverage, it supports higher data rates compared to NB-IoT and can handle voice and mobility. This means devices using LTE-M can move between cell areas (like a vehicle tracking system) or even support voice functionalities. LTE-M uses a wider bandwidth than NB-IoT, leading to the higher data rates, but this also means its devices are typically more complex and might be slightly more expensive.

The technical characteristics of those network are summarized in table 2.3.

2.3.4 . Agricultural Application

LPWANs are the best choice to be used in agricultural applications, as previously mentioned. Comprehensive surveys on the applications of LPWANs for agriculture have been presented by the authors in [232, 239, 240]. In this section we will showcase some typical agricultural application of WSNs with various LPWAN technology to highlight the advantages and disadvantages of the different protocols.

- **Sigfox:** Sigfox has been used as a monitoring environment platform, as presented by the authors in [241]. The industry widely implements

Feature	LoRa	Sigfox	NB-IoT	LTE-M
Bandwith (Europe)	868 MHz	868 MHz	180 KHz 3GPP Licensed	14 MHz 3GPP Licensed
Battery life time	8-10 years	7-8 years	7-8 years	5 years
Modulation	SS Chirp	GFSK/DBPSK	UNB/GFSK/BPSK	QPSK/16
Data rate	250bps-50 kbps	100-600 bps	66 kbps - 159 kbps	1 Mbps - 7 Mbps
Range	2-50 km	3-10 km	1-5 km	10 km
Private network support	Yes	No	No	No

Table 2.3: Main LPWANs technical specificities

Sigfox, as seen with companies like Sencrop [242], which builds agricultural sensors, including weather stations, connected through Sigfox. Another interesting application was implemented for individual cattle position tracking in mountain pastures [243]. Despite its effective implementation, limitations of the Sigfox network for agricultural applications mentioned in the literature are numerous. Firstly, the coverage of the Sigfox network solely depends on a national operator. If the Sigfox network is unavailable in a region, no Sigfox-based IoT platform can be deployed. Secondly, the usage of the ISM band limits the number of messages a device can send per day to 140. While this may be sufficient for some environmental applications, the authors in [243] pointed out that it was insufficient for tracking applications or more complex data transfer such as images. Finally, while a Sigfox node can perform uplink (sending a message from a device to the gateway) efficiently, downlink is limited (sending a message from the gateway to the device). This means that it can be challenging to confirm if a message was successfully transmitted. This not only limits actuator commands, such as opening an irrigation valve, but also poses security issues, as firmware updates over the air are not possible. Due to its proprietary nature, the Sigfox protocol has seen limited adoption in the research field. Additionally, it's worth noting that the Sigfox company has faced significant financial challenges and was acquired in early 2023, casting uncertainty on the protocol's future [244].

- **NB-IoT:** NB-IoT has been widely used in agriculture, primarily due to its compatibility with existing LTE networks that are readily available, even in rural areas. This means that farms with coverage can begin deploying sensors immediately. NB-IoT has been successfully implemented for irrigation and fertilization management projects as in [245], greenhouse environment monitoring and automation [246, 247], irrigation management [248], and aquaculture monitoring and automation [249]. Additionally, NB-IoT provides the capability to transmit larger data files,

such as images. However, this usage should remain occasional to avoid excessively draining the battery [250]. Streaming, in any case, is not possible. The limitations of NB-IoT stem from its use of licensed bands. This incurs additional costs for farmers, as they must invest in dedicated data plans, and it also ties them to the Internet Service Provider (ISP) responsible for network coverage in the area. Furthermore, many farms worldwide, especially in developing countries, are too rural and remote to receive LTE coverage.

- **LTE-M:** LTE-M is viewed as the most suitable LPWAN technology for IoT applications requiring high data rates, low latency, and full mobility (i.e., where sensors are mobile). It also provides the highest density among LPWANs, supporting up to 1 million devices per km². In agriculture, there aren't many research implementations. The authors in [251] proposed an image acquisition system for field monitoring. The specific features of LTE-M often make it seem overkill for agricultural applications. The trade-off between cost, data rates and energy consumption is less appealing from an application perspective, as discussed by the authors in [232]. However, for specific high data rate applications, such as image transfer, it can be valuable. As a result, researchers have suggested a heterogeneous LPWAN network that uses LoRa for sensor monitoring and LTE-M for base station communication like in [252]. LTE-M also has the same issue as Sigfox and NB-IoT which is the network coverage dependency of a national operator.
- **LoRa:** LoRa has been widely adopted for agricultural projects and has been the focus of dedicated studies [253, 254]. Network experiments conducted by the authors in [255] further underscore the protocol's suitability for agricultural applications. LoRaWAN WSNs are incorporated in every agricultural application cited in section 8, including greenhouse monitoring and automation [256], crop yield monitoring [257], and irrigation management [258]. In certain conditions, LoRaWAN even supports image transmission, as successfully demonstrated by [259] in agricultural contexts. Lastly, LoRa proves effective for livestock monitoring [260], given that LoRa nodes facilitate mobility [261] and even offer basic localization through gateway triangulation [262].

From the literature, LoRaWAN emerges as the most suitable protocol for our application context, as indicated by the authors in [255]. The rationale is threefold. Firstly, LoRaWAN fulfills the coverage and energy consumption requirements for farm deployments. Secondly, since LoRaWAN operates in an unlicensed spectrum with an open-source protocol stack, farmers can deploy a LoRa network on their farms both easily and cost-effectively. In contrast, Sigfox, NB-IoT, LTE-M, and other LPWANs necessitate infrastructure de-

ployment by their respective operators. This is particularly crucial for small-scale farms in developing countries, which are predominantly situated in rural and remote locations with restricted access to existing network infrastructure. Lastly, the open-source nature of the LoRaWAN protocol stack empowers developers and researchers to suggest improvements for various use cases and ensures maintenance support from the community.

2.3.5 . Limitations

LPWANs are designed for wireless communication with a focus on long-range and power efficiency, but this comes at the cost of low data rates. LPWAN have multiple limitation as explained by the author in [234]. This low data rate not only makes them unsuitable for real-time or high-bandwidth applications, such as video streaming, but also hinders the transfer of large data files, such as images or sound, typically used by machine learning algorithms in the cloud. The latency in LPWANs can be high, in part because the data payloads for individual messages are limited but also due to the duty cycle imposed in certain region due to government regulation, making LPWAN unsuitable for applications that require real-time responses. This regulation often impose that a device is allowed to communicate a fraction of the time (usually 1% to 10%). The duty-cycle regulation are particularly constraining for LPWAN using unlicensed frequency band like ISM band. Moreover, LPWAN doesn't provide high-Quality of Service features, so there's no guarantee of consistent data delivery or message ordering. In environments with many devices, especially on unlicensed spectrums like LoRa, interference can become a concern. From a security standpoint, while LPWAN technologies incorporate safety features, the limited capabilities of LPWAN devices can constrain the robustness of security measures. Geographical and physical obstacles can sometimes challenge the advertised extensive coverage. Also, devices using LPWAN often lack powerful processing capabilities due to the focus on power efficiency and low cost. The LPWAN market has multiple competing technologies, which can lead to fragmentation and challenges in interoperability. Some LPWAN types, especially those based on cellular systems, also require existing network infrastructure to operate effectively. However, despite these limitations, LPWAN remains invaluable for many IoT scenarios that prioritize range and battery life over data speed.

2.3.6 . Amelioration axes

LPWANs are particularly suited for telemetry sensors in agriculture, such as temperature, humidity, wind speed, etc. However, more complex sensors, such as cameras or microphones, struggle to send their data over such limited networks. This becomes a significant issue as an increasing number of agricultural IoT applications rely on this kind of data, as shown in 8, especially with the assistance of machine learning algorithms performed in the

cloud, as discussed in section 2.2.2. Thankfully, a new technological paradigm, called Embedded Artificial Intelligence, aims to shift the computation process directly to the end devices. This can also be called Edge Computing. This shift is becoming increasingly feasible due to the advancement of Embedded AI methods and the cost reduction of high-performance microcontrollers, like 32-bit microcontrollers. Using these methods, a sensor can analyze complex data directly on the device, sending only the inferred result to the application server in a concise message via LPWAN, thereby conserving energy. However, Embedded Artificial Intelligence requires the capability to occasionally offload its complex data and to be updated remotely. These needs, along with other characteristics of Embedded AI, will be explained in greater depth in the next section 2.4. For LPWANs and in particular LoRa as it is the best suited LPWAN for agricultural application, these requirements indicate the need to develop new protocols to ensure the proper operation of Embedded AI sensors. Addressing this need while being energy efficient will be the primary focus of this thesis regarding the network contribution.

2.4 . Embedded intelligence

2.4.1 . Definition

To grasp the concept of Embedded Intelligence more effectively, it's essential first to outline the computational architecture of an IoT network [78]. The computational processes to exploit and interpret the perceived data within an IoT platform can occur at multiple locations across the network. Typically, these processes are executed at the Edge, within the Fog, or in the Cloud. We talk about edge computing to describe the processing of data at the source of data generation, often at the edge of the network [263]. This can be on devices like IoT sensors, gateways, or other equipment constituting a point of collection. The primary advantage of edge computing is that it reduces the need to send vast amounts of data across a network, leading to reduced latency, bandwidth usage, and potential points of failure. Fog computing is an architectural approach that extends the concept of edge computing. It decentralizes computing resources and operations, distributing them closer to the data sources but often at a layer above individual edge devices [264]. Fog computing is similar as an intermediary layer between the edge and the cloud. It aims to improve efficiency, reduce data transfer loads, and provide quicker data processing and analysis closer to the source. It often involves aggregating data from multiple edge devices and then processing it in a local gateway or other local computing resource like a server on the local network. Fog computing can also refer to distributed computation approach where the computation process of data perceived by one edge device is distributed amongst other devices on the same network. Fog computing may vary in definition de-

pending on research approach but we define it as processing data within the local network without accessing the internet. Finally, cloud computing refers to the delivery of various services over the internet, including storage, processing, and software solutions [265]. These services are delivered from data centers that can be located anywhere globally, providing immense computing power and storage capacity.

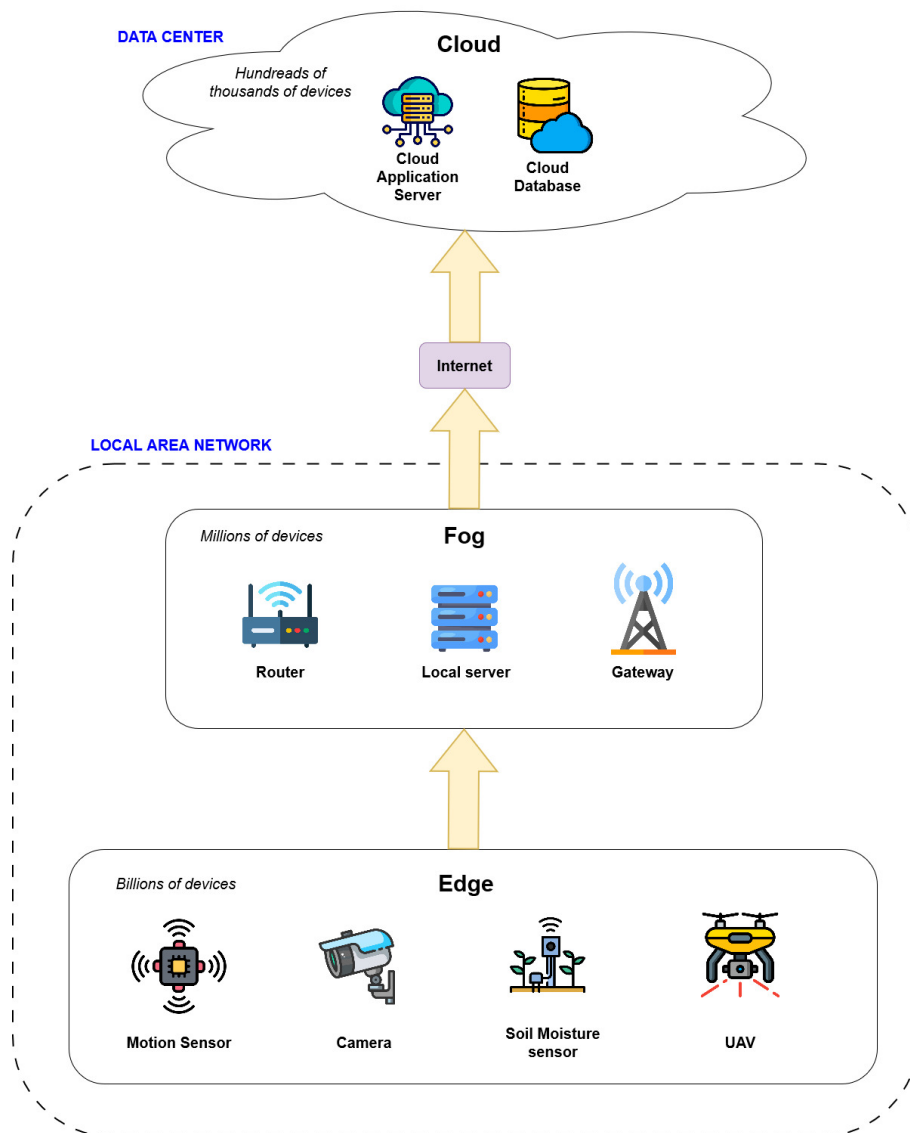


Figure 2.10: Cloud, Fog, Edge data processing architecture

The relationship between edge, fog, and cloud computing can be visualized as a hierarchy or continuum of data processing layers depicted in figure 2.10 and is discussed more in depth by the authors in [266]. The Edge Layer

is at the bottom of this hierarchy. Individual devices (like sensors or smart cameras) process data on-site before deciding what data to send upwards. As we move up, fog nodes or gateways might collect data from several edge devices, performing preliminary processing, filtering, or analysis. At the top, the cloud receives data that might have been aggregated and pre-processed by the lower layers, conducting more extensive analysis, storage, or leveraging its vast computational resources for tasks that aren't time-sensitive or that require significant computational power like Artificial Intelligence. The overall idea of this layered architecture is to avoid any unnecessary data transmission which are energy costly, weight on the network bandwidth, prone to error and security failure and increase latency. Together, these layers provide a multi-tiered approach to data processing, allowing for efficient, scalable, and responsive IoT systems, ensuring data is processed in the most suitable location based on latency, bandwidth, energy, and computational requirements.

Usually, processing data at the edge is preferable and can lead to faster decision-making due to reduced latency, as data doesn't need to be transmitted and then received from a centralized system. Moreover, this approach can increase privacy and security since data remains on the device, reducing transmission-related security risks. Devices can also continue to function even when offline, ensuring operational continuity [263]. Finally, in the context of WSNs, the energy required for wireless transmission often represents the most significant energy expense for a device [221]. Therefore it is preferable to limit wireless communication to expend battery lifetime.

Unfortunately, complex algorithms, such as AI models, require considerable computational power to execute and most of IoT device possess limited computational capacities. However, AI-driven data analysis is becoming increasingly prevalent, especially for intricate data processing tasks involving images or audio. In the agricultural realm, camera sensors are of utmost importance. For instance, they can leverage computer vision AI algorithms to detect if a plant is manifesting a disease or to monitor livestock, as highlighted in section 8. Regrettably, we also deduced in section 2.3 that devices in agricultural WSNs use LPWAN technology to facilitate long-distance communication with minimal energy consumption, but this comes at the expense of data-rate. The entire procedure of transferring images for AI analysis from a large number of nodes to the cloud within agricultural WSNs is consequently time-consuming and can cause network congestion, significantly impacting battery life of node devices.

To address this issue, researchers have sought to shift the AI computational process directly to the end device developing techniques in a new field known as Embedded AI. Embedded AI integrates artificial intelligence capabilities directly into edge devices, allowing them to process and analyze data locally rather than relying on cloud-based systems for these computations.

To do so despite the limited computational power of typical IoT devices as explained in 2.2.2, and without increasing costs too drastically, two approaches are explored, the hardware one and the software one as discussed in [198]:

- The hardware approach involves designing and developing specialized computer chips or logical component, that are optimized for running AI algorithms. such dedicated purpose integrated circuits are to opposed to generic purpose ones such as Microcontroller and Microprocessors . Dedicated hardware can provide substantial performance gains over traditional chips architecture, allowing for real-time processing, lower power consumption, and support for more complex models. Different type of hardware approach exists:
 - **Graphical Processing Units (GPU):** GPUs are specialized electronic circuits designed to accelerate the processing of images and videos to be displayed on a computer's screen. Unlike microprocessors, which handle general-purpose tasks in a computer, GPUs are optimized for parallel processing, making them highly efficient for tasks that involve large-scale data computations, such as graphics rendering and deep learning applications. Over time, the high parallel processing capabilities of GPUs have led to their use not just in graphics-related tasks but also in various computational workloads, especially in the fields of artificial intelligence and scientific simulations, as explained by the authors in [267]. In the realm of IoT devices, GPUs can either be used in combination with a microprocessor or a microcontroller. When combined with a microprocessor, we refer to the device as a System on Chip (SoC). A SoC is an integrated circuit that consolidates multiple components of a computer or other electronic system into a single chip, encompassing a CPU, memory, input/output ports, and often other specialized hardware. Well-known examples of SoCs include the Raspberry Pi and Nvidia Jetson Nano. The Jetson Nano is a particularly well-optimized SoC for embedded AI applications, as shown by the authors in [268]. SoCs can sometimes be too costly for certain applications, and in such cases, GPUs can be combined with microcontrollers. Authors in [269] have successfully implemented a road traffic sign detection system using a combination of a microcontroller and GPU. However, GPUs typically require extended RAM capacity and higher energy consumption, making them less suitable for basic microcontrollers or battery-powered devices, as discussed in [270].
 - **Application-Specific Integrated Circuit (ASIC)** An ASIC is a custom-designed chip optimized for a specific application or function, rather

than for general-purpose use. In this context, GPUs can be considered ASICs for graphical processing. Although GPUs have been widely repurposed for AI applications, dedicated integrated circuits for AI have been engineered, as shown by the authors in [271]. For example, Google has developed an ASIC for running the famous TensorFlow algorithm, called Tensor Processing Units or TPUs [272]. Other intriguing approaches include Neural Processing Units (NPU) dedicated to running neural networks and Visual Processing Units (VPU) as mentioned in [273]. While most of these ASICs are geared towards server applications, there have been research to embed such components into IoT devices as explained in [274].

- **Field Programmable Gate Arrays (FPGA)** ASICs, including GPUs, are not purely general-purpose hardware; they're designed to run specific types of algorithms within their domain of application. However, hardware can be tailored to meet the unique needs of particular algorithms through the use of programmable logical units, among which FPGAs are the most prevalent. FPGAs are integrated circuits that can be reconfigured post-manufacture to implement various digital circuits. Unlike ASICs, which are permanently designed to perform a specific function, FPGAs allow designers to program and adjust the hardware configuration to suit various tasks and applications, particularly AI acceleration, as described by the authors in [275]. They are especially well-suited for embedded AI applications, as explained by the authors in [276], because they can be easily reprogrammed. However, their cost can be a significant drawback for large-scale device deployment.
- The software approach, on the other hand, primarily involves optimizing and adjusting existing AI algorithms and models so they can run efficiently on hardware-constrained devices such as microcontrollers. There's a balance to strike: excessive simplification could result in a loss of model accuracy or performance. Moreover, even optimized models might not perform adequately on extremely resource-constrained devices. To optimize AI algorithms effectively, we must deepen our understanding of the domain. AI is a multidisciplinary field of computer science focused on creating systems capable of performing tasks that would ordinarily require human intelligence. These tasks encompass problem-solving, understanding natural language, recognizing patterns, making decisions, and interpreting complex data, among others. Due to the multidisciplinary nature of AI, there exists a vast variety of methods, approaches, and algorithms for different applications. The authors

in [277] have provided an excellent survey on this topic. It appears that within the realm of AI, the most prevalent domain is Machine Learning, and among Machine Learning techniques, the most commonly used is Neural Networks (NN). Neural networks mimic the functioning of the human brain by utilizing an extensive web of interconnected processing units called neurons. Those neurons forms layers, and when there is more than one layer, NNs are often referred to as Deep learning methods. NNs excel at identifying patterns and play a crucial role in tasks such as audio pattern treatment, image analysis, and speech and language processing [278]. For these reasons, neural networks appear to be at the heart of Embedded Machine Learning methods, as illustrated by the authors in [38]. In the agricultural sector, we've demonstrated that sensor applications requiring AI processing at the edge typically involve image or audio file analysis 2.2.2. Therefore, NNs are also of utmost importance for embedded agricultural applications, as highlighted by the authors in [279]. The prevalence of Machine Learning and NN in the Embedded AI domain is so important that Embedded AI is often referred to as Tiny Machine Learning [280].

To grasp how to optimize NNs and embed them into IoT devices, we first need a brief understanding of how neural networks function: Neural networks operate by processing input data through layers of interconnected nodes or "neurons". Each connection has an associated weight, which is adjusted during training. When data is fed into the network, it undergoes a series of transformations within these neurons. Neurons compute a weighted sum of their inputs and apply an activation function to produce an output. As data flows from the input layer to the output layer, they go through a number of intermediate neurons forming hidden layers and the network makes predictions based on the current weights of those neurons. This process is pictured in figure 2.11.

Now that we know how NNs operate, we need to understand how to train them. The training of a neural network involves repeated forward and backward computations through each layer of the network until the desired accuracy of the model is reached. This training process can be divided in three parts and is depicted in Figure 2.12

1. *Forward Propagation* Forward propagation in neural networks involves passing the input data through the network to produce an output. Starting at the input layer, each neuron computes a weighted sum of its inputs and applies an activation function to generate an output. This output is then used as input for the next layer of neurons. This process continues through each layer of the network until the final output layer is reached, producing

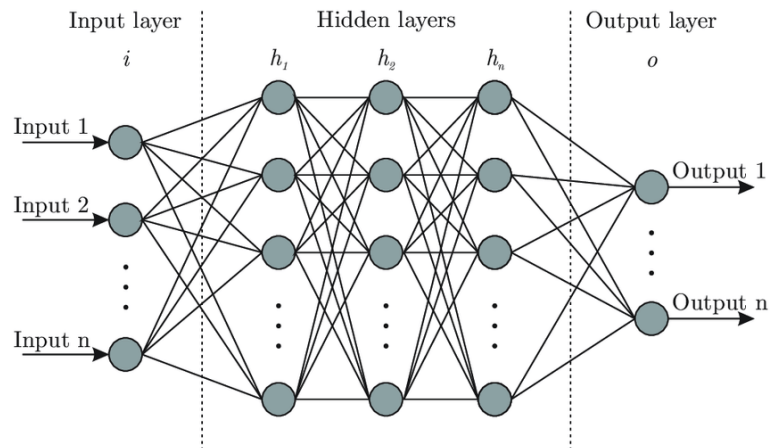


Figure 2.11: Neural Network architecture

the network's prediction. The layers are linked by the network's parameters, depicted by arrows in Figure 2.12, which consist of weights and biases. Weights dictate the significance of each input, whereas biases influence the propensity of a neuron to activate or fire. During the first training cycle of a NN, those parameters are initialized with non-zero random values.

2. *Loss function*: The forward propagation produces a predicted value, which is compared to the real (ground-truth) value using a loss function. The loss function calculates a score, or error, based on the difference between these values. This score guides parameter updates during the backward propagation process. Ideally, the loss function approaches zero, indicating that predictions closely match the real values. There is a vast amount of available loss functions with different advantages and drawbacks and those are listed by the Keras API, a famous tool for NN building [282].
3. *Backward propagation*: After forward propagation determines the prediction error using a loss function, backward propagation adjusts the network's weights and biases to reduce this error. It calculates the gradient of the loss function with respect to each weight by applying the chain rule, determining how much each weight contributed to the error. The gradients indicate the direction and magnitude of changes required. Once gradients are computed, weights and biases are updated using optimization techniques, often gradient descent (but many other optimizer algorithms are available [278]), to make the network's predictions more accurate in subsequent forward passes.

It is important to note that running a NN differs from training one, with

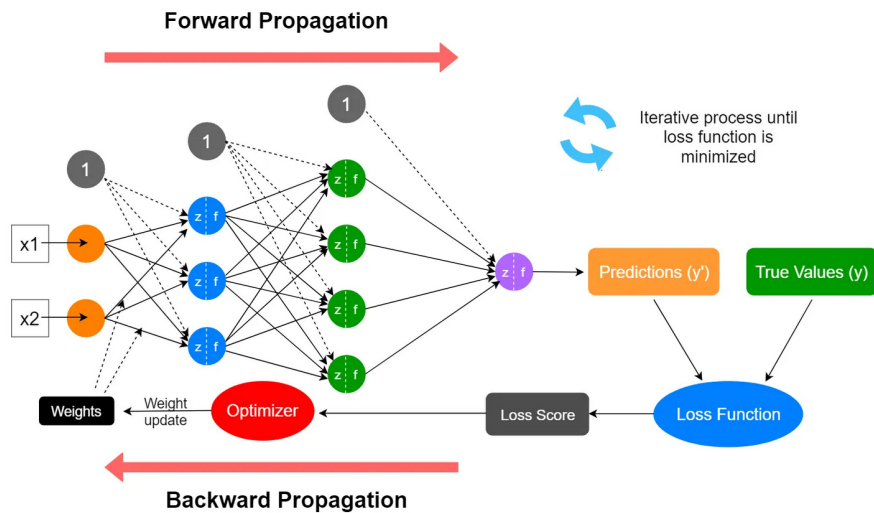


Figure 2.12: Neural Network training process [281]

the latter being more resource-intensive. Hence, Embedded AI software strategies focus on adapting pre-trained NNs for constrained hardware, delegating the training to more powerful servers, usually in the cloud [280]. For example the famous YOLO (You Only Look Once) algorithms, widely use in object detection, has been the subject of many research to simplify it and port it into constrained devices [283].

For deploying pre-trained neural networks on resource-limited devices, several strategies exist, as detailed in [284]. Presently, the predominant and most effective approach is model compression techniques. Model compression offers a suite of tools designed to lighten the computational process of the neural network. A detailed exploration of each method's workings would necessitate an extensive discussion on neural network operations, which is beyond the scope of this thesis. However, the primary tools utilized include:

- *Network structure redesign*: It involves enhancing existing neural networks by crafting novel architectural configurations. For example MobilNet [285] is an NN architecture designed specifically for mobile and embedded vision applications. It is optimized for performance on devices with limited computational and memory resources.
- *Quantization*: This refers to the compression of floating-point data bits in parameters to reduce the complexity and size of the model by simply reducing the number of bits. This method comes at the cost of model accuracy but several methods offer interesting

trade-off as shown in [286].

- *Pruning*: It refers to the technique of reducing the size and computational complexity of a neural network by removing certain neurons or connections that contribute minimally to the network's performance. The primary motivation behind pruning is to produce a leaner and faster model that can operate efficiently on devices with limited computational resources, such as IoT devices or mobile phones, without significantly compromising accuracy. By eliminating these redundant or less important neurons and connections, the neural network becomes more lightweight, consumes less memory, and requires fewer computations, making it more suitable for deployment on embedded systems. Multiple pruning methods exist and are described in this survey [287]. A simple visualized way to represent pruning is given in figure 2.13

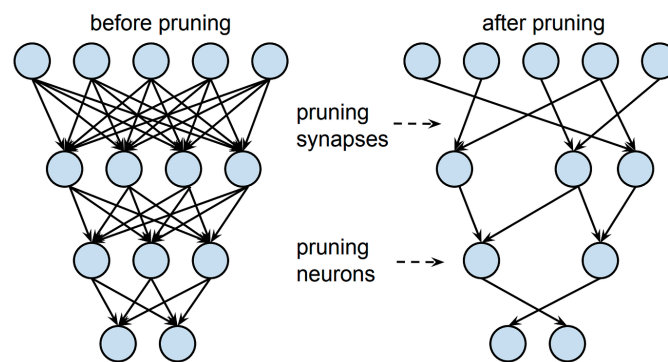


Figure 2.13: Neurons pruning process

While on-device training might not be as common as deploying pre-trained models, it presents a compelling opportunity, as discussed by the authors in [288]. The primary benefit for devices include the capability to learn without an Internet connection. Additionally, there's no requirement to send data to the cloud or retrieve an updated model, which conserves bandwidth, minimizes latency, and saves energy. Looking at it from a data viewpoint, on-device training inherently safeguards privacy. Finally, with on device learning devices can become more intelligent, handling model drift issues and updating deployed pre-trained models to better suit the environment and even individual users. For instance, a medical instrument could gradually adapt to deliver tailored predictions or services, catering to a particular patient's unique circumstances. Despite these advantages, the hardware implementation of on-device learning still demands significant processing power, making

it unsuitable for simpler microcontrollers, although there are promising results on SoCs [289].

In essence, the software approach seeks to adapt the AI to the device, while the hardware approach adapts the device to the AI. In many modern embedded systems, a combination of both approaches is used to achieve the best performance, ensuring that AI algorithms are both optimized for efficiency and have the dedicated hardware support to execute them effectively. In both case the need for model update and adaption to the specific condition where the devices is being deployed is of utmost importance, whereas it comes from on device training or remote-server induced model update.

2.4.2 . Application

Due to its multidisciplinary nature, Embedded AI has been applied to a wide range of domains, and comprehensive surveys on common applications have been conducted by authors in [290, 291, 280, 292]. Originally, it was considered that the realm of embedded AI saw its first significant contribution in applications related to voice assistants and wake words [293]. Wake words are specific words or phrases used to activate voice-activated AI systems. Examples include "Hey Siri" for Apple, "OK Google" for Google Assistant, and "Alexa" for Amazon. These words allow hands-free interaction, with the device only sending audio to servers for further analysis after the wake word is detected. This addresses some privacy concerns and limits the energy consumption of battery-powered devices such as smartphones. The intent was for a low-power microcontroller to always be listening, while the main processor is in sleep mode, avoiding excessive energy consumption by analyzing everything it hears when it's not relevant. The sole goal was detecting those words to wake the processor.

Another critically important application is in healthcare, for instance, within the cardiac domain. Wearable devices can monitor a patient's cardiac rhythm and analyze it directly at the edge without relying on the cloud. This means cardiac anomalies can be detected in real time, with appropriate measures taken, as explained in [294]. Another intriguing application in healthcare involves a system designed to detect and manage unwanted episodes in patients with Parkinson's syndrome [295].

A domain where embedded AI shows promising applications is predictive maintenance. Predictive maintenance involves detecting the conditions of machine failure and thereby averting it with easier maintenance operations compared to complex repair procedures. Authors in [296] proposed a system based on ESP 32 and a thermal camera to detect heat anomalies in machines in real time. Similarly, to detect failures in washing machines, a vibration sensor and Arduino Nano 33 BLE were employed by authors in [297], combined with the TensorFlow Lite ecosystem.

In the agricultural sector, TinyML also presents various opportunities. It's especially beneficial for small-scale farmers, providing a significant improvement in crop monitoring at minimal costs. An example is the PlanVillage project, aiming to assist African farmers in integrating IoT and Embedded AI into their farming [298]. A prominent application of Embedded AI is crop monitoring. Authors in [116] proposed an edge computing system to oversee the progress of mildew diseases in vineyards based on environmental data. Other researchers have employed simple cameras to monitor water stress in crops [299]. Similarly, cameras have been used for grape leaf disease detection [300], coffee plant leaf disease detection [301], and fruits and vegetables identification as in [302]. Additional interesting research has focused on irrigation management, like in [303], where a system for agro-environmental management using moisture sensors and real-time video analysis of soil photographs was proposed. Embedded AI can also upgrade older systems, as illustrated by a TinyML device that uses a camera to collect data from numerical water meters to monitor central pivot irrigation. Furthermore, Embedded AI has been employed in supervising atmospheric conditions in greenhouses. For instance, authors in [304] developed a cost-effective gas monitoring system potentially used to detect disease development conditions and act on them. Livestock management can also benefit from advancements in embedded AI, like the study by authors in [305] who focused on a real-time animal behavior recognition system using inertial measurement sensors in collars. Embedded AI can also assist in wildlife observation and monitoring through audio analysis, as discussed in [306].

These applications highlight the profound advancements embedded AI can offer to IoT platforms, especially in the realm of agricultural applications. However, during our research — and as noted by other authors in their surveys [292] — it was observed that TinyML implementations in specific scenarios, especially in agriculture, often overlook their network specifics. To our understanding, applications that combine the practical use of LPWAN, the most prevalent type of network in agriculture, with embedded AI, remain an uncharted territory.

2.4.3 . Limitations

Embedded AI, though groundbreaking and influential, is not without its challenges. First and foremost, the resource constraints of the device can limit the accuracy of AI algorithm outputs. Consequently, embedded AI should not be employed for critical operations where utmost accuracy is essential. Latency can also be an issue: If a model requires an excessive amount of time to execute on a device, depending on the time constraint, it might be more efficient to run the algorithms on a powerful server. Energy consumption is another nuanced consideration. By refraining from transmitting data, a device

can significantly conserve energy, which is paramount for battery-powered devices. However, if an algorithm's runtime is extended due to intricate data processing or an unoptimized embedded AI algorithm, the energy consumption could surpass that of just transmitting the data. For instance, if a device consumes 50 mW of power for transmission and 20 mW during operation, and the data to analyze is 1 kb in size with a communication speed of 1 kbps, any algorithm taking longer than 2.5 seconds to execute would make it more efficient to transmit the data directly to the server. Therefore, both network configuration and embedded AI algorithms must be meticulously assessed for each application to ensure energy efficiency.

Finally, the principal limitation of embedded AI is model drift. This concept refers to the change in the patterns of the data over time, which can affect the performance of a trained model. In other words, the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. For those reasons, it's essential to continuously monitor model performance in deployed environments to detect signs of drift. Upon detecting drift, it's advisable to retrain the model using recent data. Various real-world elements can cause data patterns to shift: evolving consumer behaviors, differential equipment wear, fluctuating economic conditions, and other such elements. Drift can manifest in various forms:

- **Sudden Drift:** An abrupt alteration in data distribution.
- **Incremental Drift:** Gradual changes over a span of time.
- **Gradual Drift:** Alternation between previous and new data concepts.
- **Cyclic Drift:** Recurring shifts to former states or patterns.

In an agricultural context, a model might be trained to enumerate strawberries in a specific fields. However, fields vary significantly in terms of strawberry type, soil color, pre-existing vegetation, and sunlight exposure for example. Therefore, once a generic fruit detection model is deployed, it needs to "calibrate" itself against gradual drift and train with local observation to enhance accuracy. Overtime the strawberry size will change in size and color introducing incremental drift. Change in season patterns will also affect the image inducing cyclic drift. Finally a sudden drift can happen if a wild pig decides to stomp into our strawberry plot.

Such drift-related challenges necessitate periodic model re-training and updates. However, as highlighted in section 2.4.1, on-device training has its limits. Presently, the most common strategy, upon drift detection in an embedded device, involves offloading data to a central server responsible for leveraging the data to train a more accurate model, which is then redeployed to the affected devices. This procedure raises network concerns as it augments overall traffic and can be energy-intensive if executed using wireless

communication protocols. Furthermore, as indicated in 2.3.6, Wireless Sensor Networks (WSN), especially in agricultural contexts, predominantly utilize LPWAN technologies. The inherent data rate constraints of LPWANs add complexity to the data offload and update process, particularly given the potential multitude of nodes.

2.4.4 . Amelioration axes

Embedded AI is still a new domain, and a wide range of solutions are currently being developed to facilitate its more widespread implementation. Solutions are available to enhance the performance of embedded AI algorithms through both software and hardware approaches. Methods have also been proposed to address the need for model training and adaptability in response to drift. One such method involves using a server to retrain model based on offloaded sensor data before uploading a new model to the devices. This method is energy-intensive and can be challenging for LPWAN networks, which are the most commonly used in agricultural scenarios. Regrettably, we couldn't find any practical implementations of embedded AI in LPWAN networks that take in account this critical need for update, even though surveys have alluded to the potential of such applications. Thus, we presume that this remains largely unexplored territory in research. Consequently, we chose to investigate the energy efficiency of embedded AI in LPWAN networks, particularly within an agricultural context, and prioritize the development of algorithmic tools for energy-efficient data offloading and model updates.

2.5 . Knowledge engineering

As explained in the introductory chapter of this thesis, an IoT platform must also proficiently interpret the data collected from various sensors. Agriculture presents a vast array of diverse data involved in the plant growing process, especially when adhering to the principles of agroecology. Therefore, to assist farmers in managing the data they have collected, efficient software tools must be provided. Such software can be very simple in certain use cases, for example, controlling an irrigation valve based solely on the soil humidity value from a sensor, with only a conditional *if-else* statement. However, such a simplistic decision-process architecture can quickly become inefficient when dealing with a vast number of cases, data, and scenarios. For instance, the number of *if-else* statements in the previous example could substantially increase if the irrigation process also took into account factors such as the type of soil, type of plants, the farm's region, future weather forecasts, time of day, etc. Therefore, it is advantageous to implement knowledge directly into a computer system, enabling it to autonomously make optimal decisions based on its understanding of a situation. To achieve this, we focused our

research on a domain of Artificial Intelligence known as Knowledge Engineering.

2.5.1 . Definition

The definition of knowledge engineering in information systems is as follows: "Knowledge engineering is the process of developing knowledge-based systems in any field, whether it be in the public or private sector, in commerce or in industry" [307]. Knowledge is also defined as "The explicit functional associations between items of information and/or data" [308]. Data refer to individual reading values from sensors, while information pertains to a preliminary degree of data analysis. For instance, considering the location and time of temperature readings (e.g., specific points within the city) allows one to discern a trend, illustrating the ongoing temperature shifts and determining a state, such as cold or warm. Finally, knowledge entails being aware of information provided by data, as well as other factors, in order to make effective decisions. The relationship between data, information, and knowledge is presented in Figure 2.14. The example illustrates that when data indicates stormy weather (an information), a person, based on their knowledge of storm conditions, content of their wardrobe, and destination, can choose an appropriate coat to avoid getting sick when leaving its place.

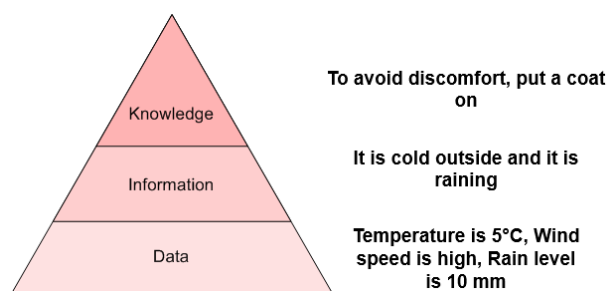


Figure 2.14: Knowledge Pyramid

Implementation of knowledge in computer system is a complex process as Knowledge can have different forms [309]:

- **Explicit:** It is a form of knowledge that can be articulated and easily passed on to others. Examples include rules from a game, maintenance procedures, or, in our context, agricultural methods. It is often documented in various media forms such as books, tutorials, videos, or other kinds of media.
- **Implicit:** Implicit knowledge refers to a type of knowledge that is challenging to articulate or share. This encompasses personal experiences, intuition, motor skills, and so forth. It is particularly relevant in the context of pattern recognition. For example, while it is straightforward to

recognize someone by their face, it becomes considerably more complex to enable another person to recognize them simply by describing their appearance.

Knowledge, being a fundamental trait of human intelligence, therefore, engineering it can be considered a field of Artificial Intelligence. Knowledge engineering refers to the conception of Knowledge-Based Systems (KBSs). There are multiple ways of implementing KBSs, as explained by the authors in [310]. For example, Neural Networks in Machine Learning are effective tools to implement implicit knowledge as they train (or in other words "gain experience") over a certain amount of data to be able to recognize patterns, such as image interpretation.

In our specific case, we aim to construct a KBS capable of making decisions regarding agroecological farm procedures based on sensor data. Consequently, the most apt software architecture for our research is known as Expert Systems, which are often referred to simply as KBSs. Expert systems, theorized since the mid-80s by the authors in [311], are software designed to generate decisions based on input and knowledge. They emulate the behavior of a human expert, meaning they can comprehend a situation (or a frame) by applying logic to it based on rules that apply to a domain of knowledge. For example, a medical system might take user input for a disease and patient characteristics, then propose an adapted posology for the treatment based on those data. Another example in smart farming would be a system capable of determining the correct amount of water for an area based on soil moisture, sun exposure, time of day or year, plant type, etc., much like a farmer would do through the knowledge acquired through education. For an expert system to function correctly, it must implement efficient tools to retrieve and model the expert's knowledge within the KB. The process of outputting a decision is termed an inference. The global architecture of an expert system is presented in Figure 2.15.

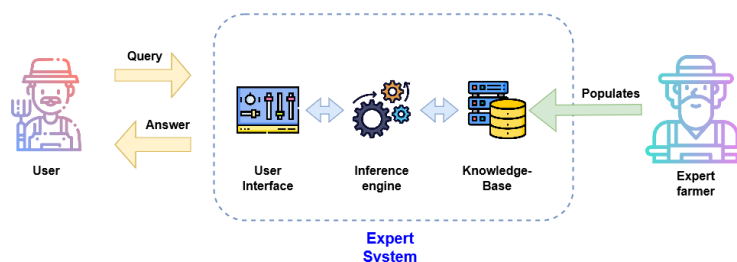


Figure 2.15: Expert System Basic Architecture

It is clear that the knowledge base (KB) is at the core of such expert systems. KBs are software models designed to represent knowledge in a form

that is accessible and interpretable by a computer. In order to be efficient, they need to have several core characteristics:

- **Formal representation:** It aims to provide a way to represent explicit knowledge. The definition given by the authors in [312] is: "A formal representation is a computer-interpretable standardized form that can serve as the basis for unambiguous descriptions of hypotheses and models in a domain of discourse." In other words, it is the structure within which the actual computer code expresses knowledge. There is a variety of tools to encode knowledge, with two of the most famous ones being RDF (Resource Description Framework) and Ontologies. They are used to describe resources and the relationships between them. Ontologies are more complex as they can allow for the representation of more intricate properties and relationships between resources. Overall, they are schematic structures representing graphs. In Chapter 4, we will explore in more depth how ontologies are implemented, as we identify them as the most suitable structure for our application. It is important to note that ontologies are so prevalent in the construction of knowledge bases that knowledge bases are often simply referred to as ontologies.
- **Scalability:** Scalability in the context of a knowledge base refers to its ability to efficiently handle a growing amount of information, users, and queries without experiencing significant degradation in performance. A scalable knowledge base should be able to expand its capacity and capabilities to accommodate increasing data and user demands while maintaining responsiveness and usability. This typically involves optimizing data storage, retrieval mechanisms, and computational resources to ensure smooth operation as the knowledge base grows.
- **Interoperability:** This refers to the system's ability to seamlessly integrate and exchange information with other systems, databases, or knowledge bases. A knowledge base with good interoperability can communicate and share data effectively with various software applications and platforms, making it possible for different systems to work together and access the knowledge within the knowledge base. This is essential for achieving a holistic and interconnected information environment, enabling data sharing and facilitating efficient collaboration across diverse tools and technologies. The idea is also to reuse already existing resources when needing access to a certain type of knowledge. For example, if I build a knowledge base for a woodworking application, it could be useful to retrieve information about trees. We could implement knowledge about trees directly in the knowledge base, but

this could take time and require expertise beyond woodworking. Therefore, if a knowledge base has already been constructed about trees by more specialized experts in this domain, it would be better to reuse that one.

It is worth mentioning that knowledge bases are also a core instrument in today's internet infrastructure. The W3C (World Wide Web Consortium), the organization in charge of promoting the technological compatibility of the web, has pushed toward the implementation of what they call the Semantic Web, which they define as "The Semantic Web provides a model that allows data to be shared and reused among multiple applications, companies, and user groups" [313]. The Semantic Web is an extension of the World Wide Web that aims to enable machines to understand the meaning of information on web pages and in digital content. It is based on the idea of adding semantic metadata to web resources, allowing computers to process and interpret the data in a more intelligent way. This metadata helps create a web of linked data with explicit relationships between concepts and entities.

In conclusion, knowledge bases are often dedicated to a specific domain and should maximize the reuse of semantic knowledge already available. In the following sections, we will highlight some specific applications of knowledge bases for IoT in agriculture and how they can help identify the missing resources for our agroecological scenario.

2.5.2 . Internet of things application

IoT has been an area where Knowledge Engineering can offer substantial benefits, particularly in ensuring interoperability. As mentioned in Section 2.2.2, one of the major limitations of IoT is the diversity of devices, protocols, and associated services. A knowledge base in this area could address this issue, and as a result, multiple efforts have been made towards the creation of such resources. In their well-crafted surveys, authors in [314, 315] have explored Semantic applications for the IoT domain comprehensively. They identify various types of knowledge bases to cater to different needs within IoT: those focusing on data management, those concentrating on hardware representation, those dedicated to services, and those emphasizing security. Hybrid approaches also exist.

The W3C has shown a keen interest in the semantic representation of IoT. Similar to the semantic web, the semantic representation of Things in the Internet of Things (IoT) could offer significant advantages to web infrastructure, especially considering the exponential deployment of IoT devices. Therefore, since 2012, they have developed the Semantic Sensor Network (SSN) with the same intention [316]. SSN takes the form of an ontology and serves the purpose of describing sensors, their observations, related procedures, studied features of interest, the samples used, observed properties, and even actu-

ators. SSN employs a modularized architecture, both horizontally and vertically, and incorporates a self-contained core ontology named SOSA (Sensor, Observation, Sample, and Actuator) to represent its fundamental classes and properties. Due to their distinct scopes and levels of axiomatization, SSN and SOSA collectively enable support for a wide range of applications and use cases. The domain of the semantic Web applied to things is often referred to as Web of Things (WoT) [317]. Nowadays, the SSN serves as the foundation for various IoT knowledge-based applications, such as wireless sensor networks [318] and cloud operations [319], among others.

2.5.3 . Agricultural Application

The creation of knowledge bases for agricultural purposes has been a focus of research in the Smart Farming Domain. Comprehensive surveys on available semantic resources and knowledge bases for agriculture have been conducted by authors in [320, 321]. Since the mid-1980s, the Food and Agricultural Organization of the United Nations has developed and maintained AGROVOC [322], which is the largest and most comprehensive semantic resource for agricultural knowledge. AGROVOC comprises over 40,000 concepts available in more than 40 languages, encompassing knowledge about plants, farming methods, forestry, nutrition, and various other areas related to agriculture. However, AGROVOC functions more as a thesaurus than an ontology, which means that the relationships between entities are more hierarchical than ontological, resulting in less extractable knowledge [323].

Managing and exploiting one large repository for all agricultural knowledge can be challenging. Hence, it is often preferable to create smaller knowledge bases on various topics and link them together. There is a wide variety of ontologies covering every aspect of agricultural procedures, types of plants, and more, which are grouped into repositories. Some notable repositories for ontologies include AgroPortal [324], The Crop Ontology [325], Planteome [326], CIARD Ring Portal [327], and GODAN (Global Open Data for Agriculture and Nutrition) [328].

What sets these repositories apart from their larger counterparts, such as AGROVOC, is their decentralized development approach. In these repositories, motivated individuals can contribute specific or specialized information that may be of interest to smaller user groups but might get overlooked by larger ontology developers. A prime example of this is CropPestO [329], an ontology model for identifying and managing plant pests and diseases. This ontology not only covers different types of threats to various crops but also provides detailed information about their treatment, conditions for development, and numerous other meta-properties. It's likely that such fine-grained information might be absent from larger ontologies.

In the specific domain of knowledge bases dedicated to organic agricul-

ture or agroecology, our research has led us to the conclusion that this is a relatively underexplored area. We have not come across any ontologies with the keywords "Agroecology" or "Organic Agriculture" in the previously mentioned repositories. Our literature review has revealed only a limited number of research works in this field: In 2009, the European Union launched Organic.Edunet [330]. The goal was to provide a tool based on ontology and semantic resources, including AGROVOC, to create a platform for educating younger generations about Agroecology and Organic Farming. In 2018, authors in [331] proposed an ontology dedicated to promoting Organic and Good Agricultural Practices in Thailand. However, the exact implementation details remain somewhat vague, and no link to the actual ontology has been provided. Since 2019, the French government has been funding a knowledge base for sharing agroecological transition knowledge. However, it's important to note that this knowledge base is primarily oriented towards providing general information to farmers rather than managing an IoT platform [332]. In 2021, researchers from INRAE (The French National Institute for Agriculture and Ecology Research) proposed a position paper from a biological perspective, emphasizing the need for ontological development in organic agriculture [333]. During the same year, researchers in the field of social sciences also highlighted the necessity of an ontological approach for Agroecology [334]. Finally, three recent research works have focused on the practical implementation of a general ontology for what they refer to as Climate Smart Agriculture [335, 336, 337]. While they offer interesting approaches to model agroecological principles, they may be insufficient for our application, particularly concerning IoT, as discussed in Chapter 4. Lastly, recent work conducted this year has utilized a knowledge base approach to implement a Deep Neural Network System Using Ontology to Recommend Organic Fertilizers for Sustainable Agriculture [338]. However, this work is dedicated to a specific use case only.

2.5.4 . IoT and agricultural Application

KBSs that focus on both agricultural applications and agriculture itself are still a relatively new area of research. The first comprehensive survey we found encompassing both domains was published in June 2023 by the authors in [339], even though the authors in [321] briefly refer to IoT applications. From these works, we identified three major IoT and agricultural Knowledge Base architectures. The first one, called AgOnt, was proposed by authors in 2010 [340]. The authors created the "AgOnt" ontology to capture relationships in the lifecycle of agricultural products, considering data such as humidity or temperature. These relationships involve product properties (e.g., location, timestamp, environmental parameters, device, processing status) and connections to source products (e.g., a plant's relationship to its seedlings from

which it originates). However, it is merely an theoretical architecture for mapping food produce according to data. The second architecture was proposed in 2016 by the authors in [341], and it focuses on linking device data and farm operations through knowledge base reasoning. They based their ontology on the Semantic Sensor Network previously mentioned in [316]. They proposed two use cases to demonstrate the efficiency of their system: one regarding cow fertility detection and management through heat sensors, and the other regarding soil fertility, considering soil composition and moisture levels. They conclude the need for a broader knowledge base for agricultural and IoT applications, as their work serves as a showcase of the potential of such applications. Finally, the authors in [342] proposed an ontology based on the SSN and other agricultural repositories to create a system capable of managing various farming actions. They successfully tested their system with irrigation management based on plant characteristics. However, once again, it is a broad model that requires knowledge acquisition. Other knowledge-based systems have been implemented with IoT for specific agricultural application. For example, the authors in [122] proposed a KBS to monitor early plant diseases based on weather data from sensors.

Overall, our research concludes that implementations of knowledge bases in the domain of IoT and agriculture are still scarce. Therefore, various types of architectures should be proposed to outline the advantages and drawbacks of each, with the aim of providing a future standard akin to the SSN. Moreover, we found no applications regarding IoT device management and agroecology. Finally, many of the KBSs studied regarding IoT and agriculture did not implement different types of sensors regarding their capacity to provide embedded intelligence.

2.5.5 . Limitations

The limitations within this domain are manifold. Firstly, the process of gathering expertise-driven knowledge can be arduous and demanding. Crafting a comprehensive ontology or knowledge base for the agricultural sector is an intricate endeavor. Agricultural knowledge is extensive and in a state of constant evolution, rendering the task of encapsulating every facet of agriculture within a single ontology or knowledge base quite challenging. Based on this observation, we wish to highlight the absence of knowledge-based systems specifically tailored to Agroecology and Organic Farming. This research gap becomes even more pronounced in the context of agroecology and IoT, where, to the best of our knowledge, no KBS has been proposed to address inquiries within this specialized field. Lastly, we must also acknowledge the limitations of integration within the emerging Embedded AI paradigm, particularly in our specific case involving TinyML and LoRa. Nevertheless, it's important to note that despite these challenges, knowledge bases hold the poten-

tial to significantly enhance the organization, retrieval, and decision-making processes in the field of agriculture.

2.5.6 . Amelioration axes

Within the field of knowledge base development, the creation of a specialized ontology designed for the management of Agroecological and IoT-based farming practices holds the promise of enhancing sustainability and aiding farmers in effectively managing their farms. Furthermore, exploring the incorporation of smart sensor characteristics and requirements within this domain could yield valuable insights that extend beyond our own, warranting careful examination.

2.6 . Conclusion and proposed contribution

This chapter aimed to provide an overview of Smart Farming research within the realm of IoT. Our exploration revealed several key limitations in various aspects of IoT platforms for agriculture. Firstly, on the networking front, the emergence of the Embedded Intelligence paradigm, particularly TinyML, offers essential capabilities to sensing devices, allowing them to operate independently of the cloud and be more energy-efficient. However, challenges related to model drift necessitate that TinyML devices can receive firmware updates and offload certain data for model retraining in the cloud. This requirement becomes particularly complex in Low-Power Wide-Area Network (LPWAN) settings, where devices communicate at low data rates and with low duty cycles, limiting the amount of transferable data per day and the size of data and firmware update transfers. Secondly, from a decision-making perspective, both IoT platforms in the agricultural domain and the IoT domain, in general, are inherently complex, involving a multitude of data sources. Efficient knowledge management techniques are required to navigate this complexity effectively. This need is particularly evident in the field of Agroecology, where environmental conditions and ecosystem behavior heavily influence the proposal of efficient farming methods based on collected data.

These limitations have directed our thesis towards two primary contributions. The first contribution aims to demonstrate the energy-saving efficiency of TinyML and LPWAN architecture compared to cloud-based models. We will utilize the LoRaWAN LPWAN network, identified as the most suitable for agricultural applications. Additionally, we will propose an efficient architecture that enables model retraining in the cloud through punctual image offloading via LoRa and Firmware Update Over the Air through LoRa. The second contribution focuses on the decision-making aspect. We propose the creation of an ontology to assist farmers in managing sensor networks in Agroecology farms. Our ontology will primarily address sensors management, especially

smart sensors equipped with TinyML, and agroecological responses concerning plant and environmental characteristics.

3 - An energy-efficient wireless sensor network for Embedded AI devices

3.1 . Introduction

Based on the review presented in Chapter 2 of this thesis, we identified that IoT platforms, especially within the agricultural sector, possess several technological limitations that need addressing. This chapter delves into the design and testing of an efficient Wireless Sensor architecture for Embedded AI devices for LPWAN agricultural network. As detailed in section 8, the application of Artificial Intelligence is central to Smart Farming. AI can predict weather and climate patterns, monitor crop growth, forecast yields, automate processes with robotic machinery, and offer actionable insights derived from vast data sets collected by various sensor devices. This leads to more sustainable and efficient farming practices. A particularly promising application of AI in agriculture is the processing of complex data such as images or sounds using computer vision or audio analysis algorithms.

Those AI algorithms usually tend to require higher computational processing power than typical IoT devices, such as microcontrollers or simple Systems On Chips, can provide. Therefore, as shown in section 2.4, power-constrained IoT devices often offload the computational processes of AI to other devices in the fog or a cloud server. This introduces multiple issues. Firstly, there are concerns about privacy and security since data being transferred are subject to security breaches or can be unethically used by the cloud service provider. Secondly, there's the matter of latency. The overall process of transmitting the data to another device can introduce significant delay, which might be detrimental, especially for time-critical applications. A third challenge arises when there is a loss of connectivity, rendering the devices useless. Lastly, and perhaps most significantly for battery-powered devices, is the issue of energy consumption. Wireless communication is the most energy-intensive operation for constrained devices and often accounts for the majority of battery usage. In the context of agricultural wireless sensor networks, this energy constraint becomes even more pronounced. Devices often need to communicate over long distances and are scattered over vast geographical areas, making battery management and recharging complex. To conserve energy during communication, a specific type of wireless network called LPWAN (Low Power Wide Area Network) is employed. These networks allow devices to communicate over long distances with minimal energy costs but come with the trade-off of reduced data rates. This lower data rate can hinder or even prevent cloud-based computer vision applications from being implemented

effectively in farming environments.

Hopefully, as shown in section 2.4, recent technological advancements have brought AI algorithms to the edge with methods like Tiny Machine Learning. This allows simple microcontrollers to run complex neural networks directly on their computation-constrained hardware. This approach does come at the expense of some accuracy, but the trade-off is usually acceptable for non-critical applications, such as most farming operations. Despite promising results, this new AI trend has a high sensitivity to model drift. This means that even if a model is trained for a generic application, it will often require adjustments and improvements once deployed in the specific ecosystem where it operates. As a result, there's a need for model retraining using local data where it's deployed. Although the prospects for running AI models are promising, on-device learning for constrained hardware is still in an early stage for traditional microcontrollers. Model retraining is therefore often conducted in the cloud, necessitating the offloading of data that might have insufficient confidence score levels for the predictions made. Once the refined model is ready, it's then redeployed to the devices. This entire process amplifies wireless communication, leading to increased network congestion and power consumption, which can be challenging for LPWAN networks to support.

In conclusion, this context highlights different limitations regarding the uses of TinyML in agricultural LPWAN networks. The questions are multiple. Can the usage of TinyML with combination of LPWAN can be more energy efficient than cloud based system? And can LPWAN meet the requirements of TinyML regarding punctual heavy data offloading and Firmware update over the air while still being energy efficient? To answer those question this chapter will propose the development of dedicated algorithms and a series of experimentation to evaluate them in specified scenarios before drawing conclusion on the feasibility and the advantages over cloud based approach of the proposed TinyML/LPWAN WSN architecture. The focus for LPWAN will be directed towards LoRaWAN networks, as it was identified as the most suitable LPWAN technology for agricultural applications in section 2.3.6.

3.2 . Architecture proposal

3.2.1 . Scenario

To assess the feasibility and efficiency of our proposed TinyML and LPWAN architecture, we examined an agricultural use case focusing on strawberry monitoring with two functional modes: normal behavior and degraded condition mode. In the normal behavior mode, an intelligent, battery-powered camera sensor, equipped with a TinyML algorithm, is stationed in a field. Its primary task is to detect and count strawberry fruits. After determining the count, the sensor sends this data to an application server in the cloud, which

hosts a decision platform, using a LoRaWAN network . Decisions are made based on the fruit count and other factors. These might include initiating fertilizer application if the fruit count is lower than expected, taking counter-measures against potential diseases or pests, detecting thefts by humans or wildlife, or sending a harvest notification if the fruits are ripe. This action directive is then passed on to the executor, which might be the farmer or an autonomous robot in order to optimize their intervention. For example, a human or robot operator will only move or relocate itself for harvest when fruit are ready. A visual representation of this use case can be found in Figure 3.1.

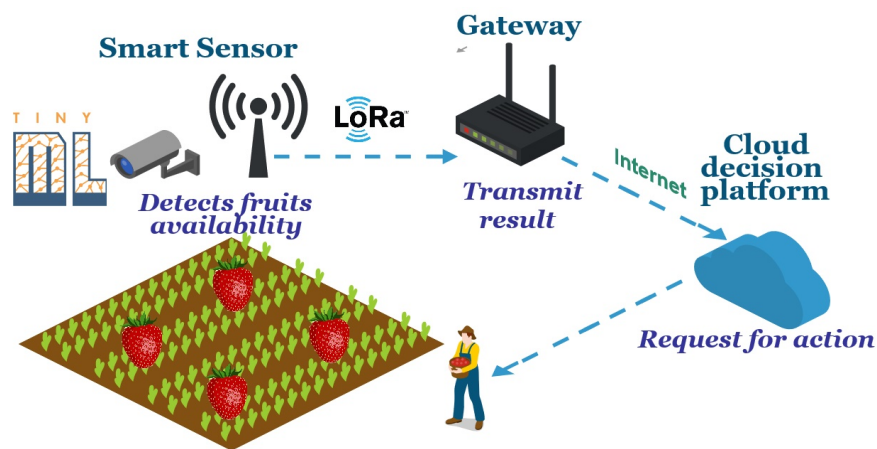


Figure 3.1: System in normal behavior mode

In the normal behavior mode, the TinyML algorithm predicts the number of available fruits with a certain confidence score. If this score drops below a specific threshold, the system shifts to the "degraded condition mode." The threshold is determined by the application's criticality. In our case, we have arbitrarily set it at 90%. In this mode, the sensor device begins offloading picture data to the LoRaWAN network without disturbing its normal functioning. This means it adapts its transfer to avoid network congestion. After collecting a sufficient amount of new data, the model is retrained in the cloud on the application server. Once this is complete, the update is transmitted from the application server to the LoRaWAN server, which then manages the Firmware Update Over the Air (FUOTA) process for the designated devices. This mode is depicted in figure 3.2.

Our system architecture is therefore made of five main components:

1. **Smart sensor:** It uses a microcontroller running a TinyML algorithm to infer the number of fruits within its field of view. If the confidence score of its inferences falls too low, it requests a firmware update over the air and offloads its data to the cloud.

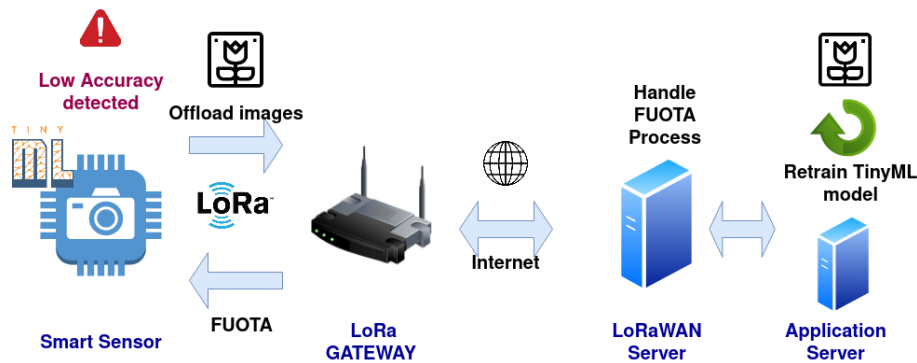


Figure 3.2: System in degraded condition mode : Data offload and FUOTA process

2. **Gateway:** Responsible for facilitating communication between the LoRa wireless physical network and the LoRaWAN server over the internet.
3. **LoRaWAN Server:** Responsible for receiving and dispatching messages between the device and the Application Server. It also oversees the FUOTA process in LoRa. The LoRaWAN server can often be located at the same location as the application server or directly on the gateway.
4. **Application Server:** It hosts the Cloud decision platform responsible for interpreting the results from the smart sensor. It is also in charge of data storage and model retraining when the system enters degraded condition mode and sufficient new data has been acquired.
5. **Performer:** The determined action is then relayed to the executor, whether it be a farmer or a robot.

From the state-of-the-art review we conducted throughout our research and through discussions with other researchers, it appears that the specific architecture of using a microcontroller to run a TinyML algorithm and communicate results through LoRa, while also being capable of updates via image transfer and Firmware Update Over The Air (FUOTA), is novel. Close works that aligns with ours are presented by the researchers in [343], who also developed a platform with a TinyML model for the communication of results over a LoRa network, specifically for mosquito wingbeat classification to count different species. Another team of research proposed a retro fit system where they use a microcontroller ESP32-cam to read the mechanical display of a water meter and communicate the data over LoRa [344]. Authors in [345] also developed a bin management system using LoRa and edge AI. However, they ran the TinyML model on a System on Chip (SoC) Raspberry Pi instead of a microcontroller and did not tackle the issue of energy consumption. All the works previously mentioned, despite demonstrating the applications of such

architectures even beyond the realm of agriculture, did not propose an update procedure to address model drift post-deployment.

3.2.2 . Environment development

To execute our proposal, we meticulously selected a range of software, hardware components, and the LPWAN communication protocol, LoRaWAN. This section will delve into each of these components in detail to elucidate their functionalities and role into our architecture.

LoRa & LoRaWAN

We will discuss the detailed workings of LoRa and LoRaWAN in this section, alongside research projects that align with our architectural objectives, specifically FUOTA and image transmission.

Among the myriad LPWAN technologies available, LoRa stands out as the most widely used and recommended for agricultural applications. This preference is attributed to its communication range, power efficiency, open-source nature, and the utilization of open frequency bands that don't necessitate expensive licenses. In practical terms, this means farmers can easily establish their own networks on their farms with the only major expense being the gateways and other hardware equipment such as server if he does not rely on the cloud for the application process. Historically speaking, LoRa is a recent technology and began in 2009 when two French engineers, Nicolas Sornin and Olivier Seller [346], wanted to develop a long-range, low-power modulation technology. They created their own startup in 2010 called Cycleo that revolved around this goal. Convinced by the potential of this long-range and low-power technology, Semtech a major semiconductor company, acquired Cycleo in May 2012. Semtech pushed the creation of a MAC protocol to help the standardisation of the technology by defining message format and the implementation of security features, this protocol will later be referred to as LoRaWAN and is distributed in open source. In February 2015 the LoRa alliance [347] was created to support and promote the global adoption of the LoRaWAN standard by ensuring the interoperability of all LoRaWAN products and technologies.

To achieve long distance and low power consumption, LoRa uses a Chirp Spread Spectrum (CSS) spread spectrum modulation technique, a technology commonly used for sonars in the maritime industry and radars in aviation [348]. This phenomenon is also found in nature among dolphins and bats to detect fish or insects. Indeed, Cycleo did not invent the CSS technology, but rather pioneered the use of this technology for data transmission. From an electronic wave modulation point of view CSS consists of changing the value of the frequency over time, increasing or decreasing the frequency over a defined time slot to encode data. The process of CSS is depicted in figure 3.3

with a LoRaWAN frame. Each frame starts with a preamble and a synchronisation process before adding the data. We can see that Chirp impulse can be either up or down and that multiple chirp can be performed per time slot.

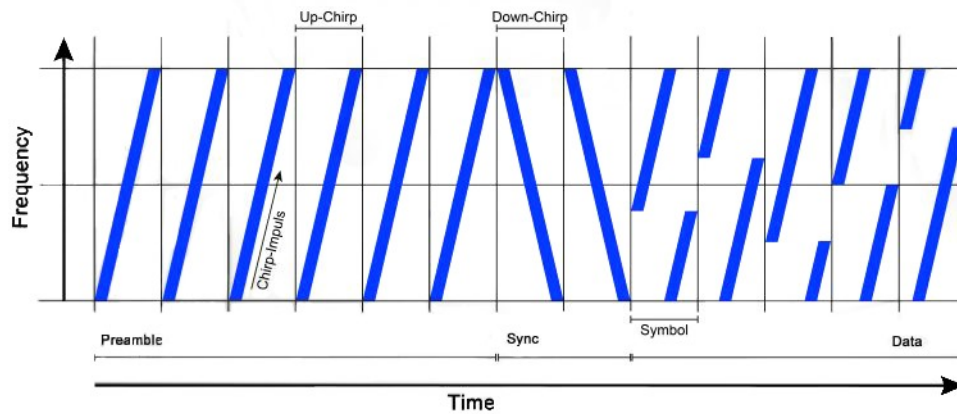


Figure 3.3: Chirp Spread Spectrum modulation of a LoRaWAN frame

LoRa technically is a proprietary technology as the transmission chips produced by Semtech is patented. However the associated Media Access Control (MAC) protocol built on top of LoRa modulation, LoRaWAN, is open source. LoRa is a bidirectional communication protocol meaning it can send uplink (from device to server) and downlink (from server to device messages) messages. A typical LoRaWAN network architecture is depicted in Figure 3.4.

LoRaWAN network are therefore composed of the following part:

- **End Devices:** Sensors or actuators transmit messages wirelessly using LoRa modulation to the gateways and similarly receive messages wirelessly from the gateways.
- **Gateways:** Receive messages from end devices and forward them to the Network Server using a a different network type called backhaul, like cellular, wifi, ethernet etc.
- **Network Server:** It primarily routes messages from the devices to the application server, verifies addresses, and ensures network availability by sending adaptive data rate commands to the devices. It is also responsible for implementing security protocols and confirming the authenticity and origin of messages. Furthermore, it oversees the acknowledgment process by relaying receive statuses to devices. If a message is received multiple times, the software sorts and retains only the relevant one. In terms of message transmission verification, LoRaWAN can be likened to a simple ALOHA protocol. It also manage downlink by choosing the best gateway for a device and also by handling the firmware update over the air processes.

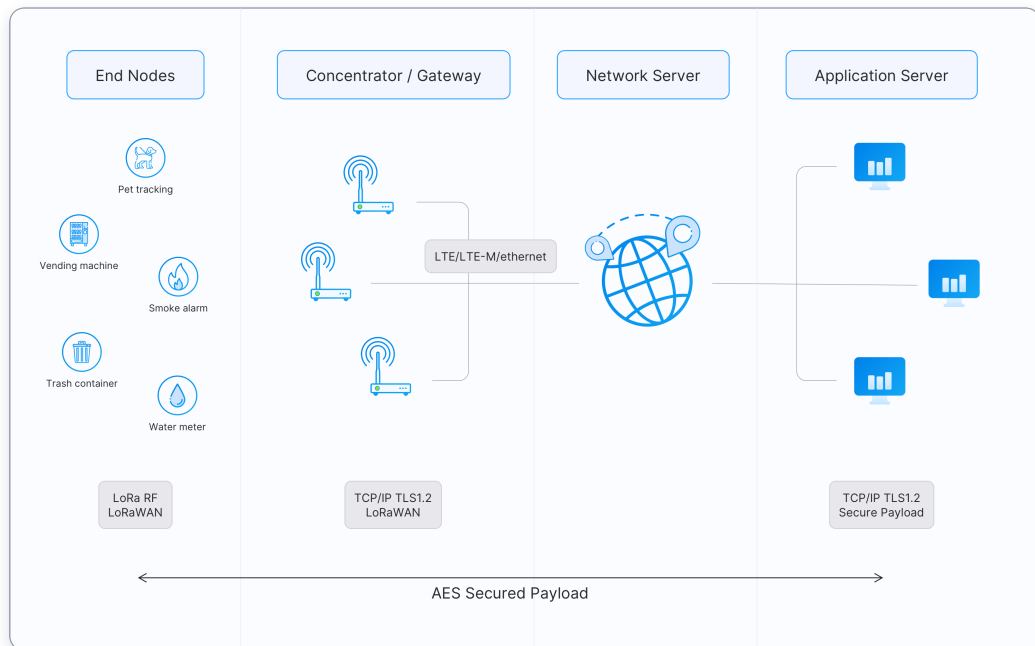


Figure 3.4: A typical LoRaWAN network architecture

- **Application servers:** Where the data are exploited.
- **Join server:** This server, that is often associated to the network server, is in charge of managing the activation of new devices in a network in a secure way.

End devices communicate with nearby gateways and each gateway is connected to the network server. LoRaWAN networks use an ALOHA based protocol, so end devices don't need to peer with specific gateways. Messages sent from end devices travel through all gateways within range. These messages are received by the Network Server. If the Network Server has received multiple copies of the same message, it keeps a single copy of the message and discards others. This is known as message deduplication.

As mention previously, LoRa frequencies falls into the category of ISM regulation. The regulation about those band variate between countries as no worldwide standard has been adopted. In Europa for example, those bands are centered around the 868 MHz frequency while in the US it is around the 915 MHz one. This ISM frequency is divided into various channels. In Europe there is a total of 16 channel with a 125 kHz bandwidth between 863 Mhz and 870 MHz. As those licenses are free to use there is still a strong regulation limitation called the duty cycle. In Europe, the duty cycle is 1%, meaning that a device can only send data during 864 second per day.

The data rate (DR) in LoRa is a crucial factor. It represents the number of

Data Rate	Configuration (SF + BW)	Bit rate (bit/s)
0	LoRa: SF12 / 125 kHz	250
1	LoRa: SF11 / 125 kHz	440
2	LoRa: SF10 / 125 kHz	980
3	LoRa: SF9 / 125 kHz	1760
4	LoRa: SF8 / 125 kHz	3125
5	LoRa: SF7 / 125 kHz	5470
6	LoRa: SF7 / 250 kHz	11000

Table 3.1: Bit rate for each data rate (DR0 - DR6) configured with the spreading factor and the bandwidth.

bits transmitted per unit of time. In LoRa modulation, the data rate is influenced by several factors, including the spreading factor (SF), bandwidth (BW), and coding rate (CR). It is expressed by the following formula:

$$DR = \frac{BW \times SF \times CR}{2^{SF}}$$

Table 3.1 displays the recommended data rate configurations and their associated bit rates, correlated with various Spreading Factor and Bandwidth values. The coding rate isn't considered as it is an external parameter dependent on the specific application. More often, it is set to its common value of 4/5.

The maximum payload of each data packet is also defined by the regulation. In Europe it is dependent on the data rate and the different values are gathered in table 3.2.

Data Rate	N
0,1,2	51
3	115
4,5,6,7	222

Table 3.2: Packet sizes (N) in bytes for each Data Rate of European regulation

Data rate core characteristics are :

- **Spreading Factor:** The spreading factor controls the chirp rate, and thus controls the speed of data transmission. Lower spreading factors mean faster chirps and therefore a higher data transmission rate.

Lower spreading factors reduce the range of LoRa transmissions, because they reduce the processing gain and increase the bit rate. Changing spreading factor allows the network to increase or decrease data rate for each end device at the cost of range. The network also uses spreading factors to control congestion. Spreading factors are orthogonal, so signals modulated with different spreading factors and transmitted on the same frequency channel at the same time do not interfere with each other. LoRa modulation has a total of 6 spreading factors from SF7 to SF12. The battery life of an end device is highly dependent on the spreading factor used. Higher spreading factors result in longer active times for the radio transceivers and, therefore, shorter battery life. The different type of SF are shown from a chirp point of view in figure 3.5

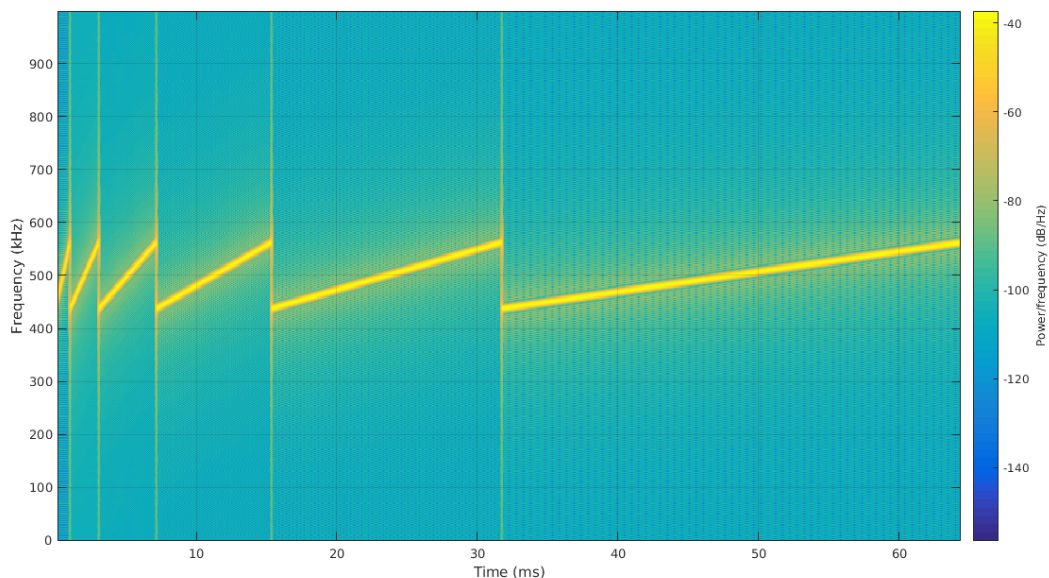


Figure 3.5: Comparison of LoRa Spreading Factors: SF7 to SF12

- **Bandwidth:** In this context, the bandwidth refers to the channel frequency utilized by the device for communication. Doubling the bandwidth leads to a corresponding doubling of the bit rate, given a constant spreading factor and coding rate. The minimum bandwidth is 125kHz, but 250 kHz or even 500 kHz can also be used, as illustrated in table 3.3.
- **Coding rate (CR):** LoRa modulation also adds a forward error correction (FEC) in every data transmission. This implementation is done by encoding 4-bit data with redundancies into 5-bit, 6-bit, 7-bit, or even 8-bit. Using this redundancy will allow the LoRa signal to endure short

Spreading factor	Bandwidth	Bit rate (kbits/s)
7	125	5.5
7	250	10.9
7	500	21.9

Table 3.3: Difference in bit rate according to bandwidth

interferences. The Coding Rate (CR) value need to be adjusted according to conditions of the channel used for data transmission. If there are too many interference in the channel, then it's recommended to increase the value of CR. However, the rise in CR value will also increase the duration for the transmission

In conclusion regarding the Data Rate, it is adaptive. This means that depending on the quality of the link within a network, which is determined based on the Signal to Noise Ratio, both the gateway and node will adjust their data rate to ensure reliable transmission. Generally, lower data rates are less susceptible to noise.

LoRa implements the simple ALOHA access protocol for channel access. Every time a device wants to use a channel, it sends its message without checking if the channel is busy. If it does not receive an acknowledgment from the receiver, it will retransmit its message after a random period of time [349].

The LoRaWAN specification defines three device types: Class A, Class B, and Class C. All LoRaWAN devices must implement Class A, whereas Class B and Class C are extensions to the specification of Class A devices. All device classes support bi-directional communication (uplink and downlink);

- **Class A:** Class A communication is always initiated by the end-device. A device can send an uplink message at any time. Once the uplink transmission is completed the device opens two short receive (downlink) windows. There is a delay between the end of the uplink transmission and the start of the receive windows (RX1 and RX2 respectively). If the network server does not respond during these two receive windows, the next downlink will be after the next uplink transmission. This shown in figure 3.6
- **Class B:** Class B devices open scheduled receive windows for receiving downlink messages from the network server. Using time-synchronized beacons transmitted by the gateway, the devices periodically open receive windows. The time between two beacons is known as the beacon period. The device opens downlink 'ping slots' at scheduled times for receiving downlink messages from the network server. Class B devices also open receive windows after sending an uplink, as you can see in Figure 3.7:

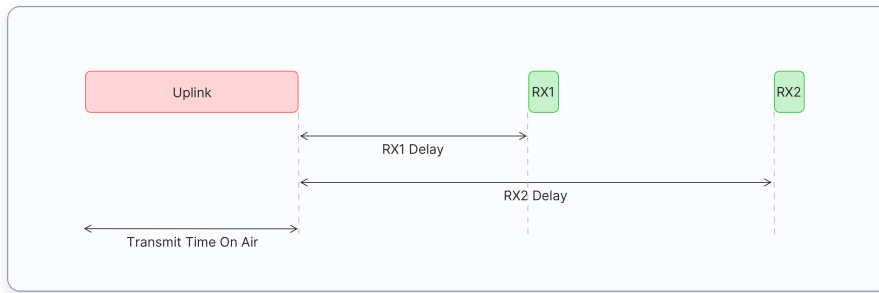


Figure 3.6: Class A receive windows

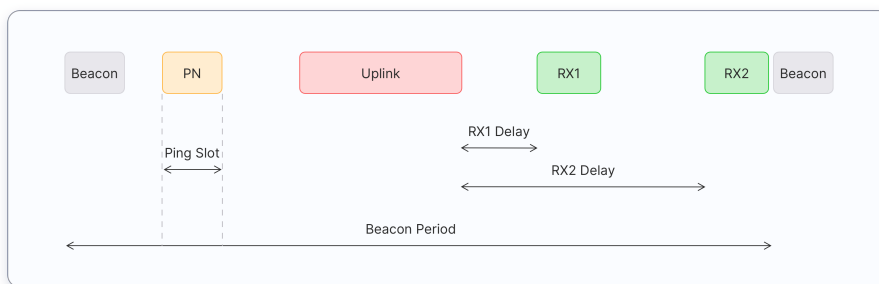


Figure 3.7: Class B receive windows

- **Class C:** Class C devices extend Class A by keeping the receive windows open unless they are transmitting, as shown in the figure below. This allows for low-latency communication but is many times more energy consuming than Class A devices. The mechanism is showed in figure 3.8.

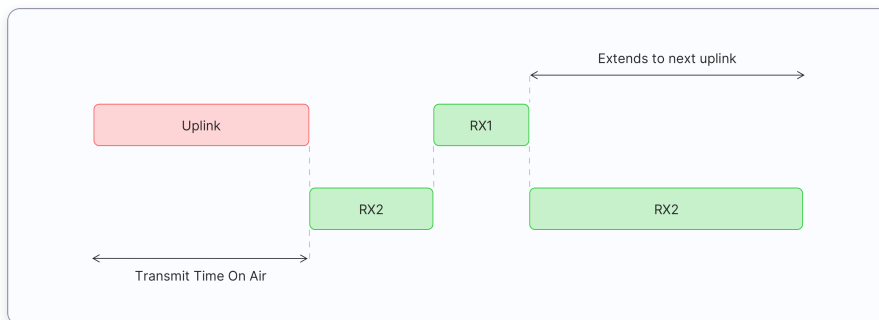


Figure 3.8: Class C receive windows

Amongst the variety of application LoRaWAN can be employed in, two of them are of special interest for us and our architecture namely the heavy media file transfer and the firmware update over the air (FUOTA).

- **Heavy data offload:** The architecture we propose that support model update for local application is, to our knowledge, novel. Consequently, we found no literature addressing this specific use case of heavy data offloading for model re-training. However, studies exploring the transfer of substantial data, such as images or audio files, using the LoRa network are available. In this section, we'll examine them to better define our strategy for implementing this functionality in our system. Firstly, we'll conduct a theoretical calculation to understand the number of pictures that can be sent daily based on the picture size. Various data rate values exist, determined by network availability, which in turn determine the speed (S) of transmission in bits per second (Bps). A lower data rate increases the probability of message transmission at the expense of bit rate. Typical data rate values for LoRa are presented in table 3.1. We are also aware of the duty cycle (DC) — the time a device can transmit data daily as per regulations. Using this information, along with the size of a media (MS), we can determine the maximum number (N) of times a media can be transmitted in a day using the formula:

$$N = \frac{S \times DC}{MS}$$

After performing the calculations, the theoretical number of media items that each data rate can send, based on media size, is shown in Figure 3.9.

Typical AI applications in agriculture demand varied data processing frequencies based on the specific application. For instance, monitoring the progression of diseases on crop leaves may necessitate data collection only once an hour. In contrast, tasks such as identifying and counting individual cows in a herd passing through a gate could require thousands of executions daily [350]. Given the diverse range of applications, it's reasonable to assume that many will need at least a hundred executions a day.

The results underscores LoRa's inefficiency in handling large file transfers, rendering it ill-suited for AI cloud data processing. Transferring files of 10 kb remains manageable, with every spreading factor able to transmit more than ten images daily. In the most optimal data rate scenarios, over a thousand can be sent. However, the system encounters difficulties with files larger than 100kb. None of the data rates can satisfy the previously mentioned requirement for agricultural applications. This also underscores the need for efficient file compression to adhere to regulatory standards, particularly when sporadically offloading large data sets.

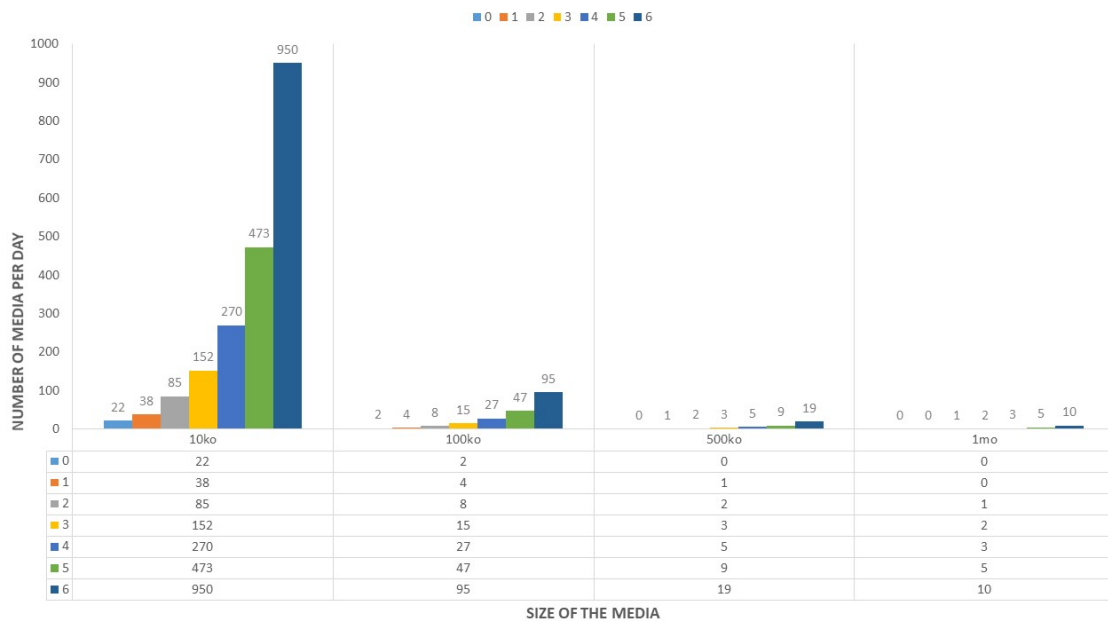


Figure 3.9: Theoretical maximum number of large media files transferable per day based on data rate and media size

Despite these limitations, LoRa has been studied as a tool for image transmission in various scenarios. From our research, we identified noteworthy projects and compiled their characteristics and limitations

- Authors in [351] proposed a method for long-distance image transmission using LoRa for mangrove monitoring. They achieved distances of up to 6 km, which was suitable for their application. For transmission, the images were compressed using the JPEG format before being converted into a hexadecimal format. Subsequently, the files were split into small packets for transmission. Although they mentioned duty cycle limitations, they did not specify the number of images transmitted per day that adhered to these constraints. The exact size of the transmitted image wasn't explicitly stated; however, an approximation of 25 kB can be inferred from their experimental results. They tested their system using various data rates, and the transmission speeds ranged from 67 to 840 seconds. This implies that between 12 and 1 image was transferred per day.
- Authors in [352, 353] tested image transmission for three simultaneous nodes using a broadcast timing algorithm. In this setup, the gateway synchronizes the transmissions of each node, allotting specific time slots for their individual image transfers. In their

experiments, they transmitted images approximately 30kb in size. They assessed the transmission of these images under various conditions and at different data rates. Under optimal network conditions (highest data rate with a clear line of sight), the transmission time was 51 seconds, while in degraded network conditions with obstructions such as buildings, it took 150 seconds. This means they could transmit between 17 and 5 images a day, depending on the conditions. The same author, Ching-Chuan Wei, also conducted an analysis of LoRa image transmission based on different encoding methods. He showed that WebP was more efficient than JPEG [354].

- Authors in [355] proposed an experimentation for LoRa image transmission over varying distances. They successfully transmitted 64 kb images over distances of 2 km in 110 seconds, 5 km in 146 seconds, and 16 km in 210 seconds. While they did not address the duty cycle issue directly, based on their results, we can infer that the maximum number of images they could transmit per day is 7 at 2 km, 5 at 5 km, and 4 at 16 km. They utilized the JPEG2000 compression technique and emphasized how image encoding can significantly influence transmission time.
- Authors in [356] attempted to implement a quasi-real-time video surveillance system that transfers LoRa frames upon detecting movement. They demonstrated that WebP is a viable compression format for images, as they were able to reduce pictures from 3.1 MB to 3.8 kB. However, during their testing, with unspecified parameters, it took up to 10 minutes to send just one picture, suggesting potential issues with their test setup.
- [357] proposed a compression and image recovery algorithm based on JPEG, where they managed to compress images that were several MB in size to between 2.7 and 27 kB. Although they didn't provide detailed transmission parameters, they successfully achieved transmission times under 30 seconds to comply with the Thing Network LoRa provider's rules.
- [358] is the first real implementation of LoRa image transmission that we encountered in the literature. The author implemented a simple model using JPEG image compression to send images of 8kB, but did not specify the actual transmission time. In subsequent work, [359], the same authors highlighted the limitations of the simple ALOHA protocol that LoRa uses for communication. They also proposed employing CSMA (Carrier Sense Multiple Access) [360] methods to allow multiple nodes to communicate im-

ages while avoiding network congestion..

- [361] proposed an image transmission system for agricultural scenarios using LoRa. They compressed images using the JPEG method and introduced a Multiple Packet LoRa (MPLR) protocol, which is between 22 and 45% faster than the traditional ALOHA method. For images ranging from 9 kb to 28 kb, transmission times varied: from 5 to 15 seconds with the best data rate, and between 49 to 164 seconds with the lowest data rate.
- Finally, authors in [259] demonstrated the use of image transmission with LoRa for agricultural applications. They proposed a system that transmits images using the JPEG 2000 compression method. While they did not specify the transmission time, they conducted tests to prove the efficiency of the transmission in terms of image integrity. Their results showed that images transferred with LoRa retained 85% of their original information during transmission.

We compiled the different papers and their characteristics in table 3.4.

Paper	Application	Compression method	Size of images	Time of transmission	Estimated maximum number of files per day
[351]	Mangrove monitoring	JPEG	25 kB	67s to 840s	1 to 12
[352, 353, 354]	Experimentation purposes	JPEG, Webp	30kB	50s to 150s	5 to 17
[355]	Experimentation purposes	JPEG2000	64 kB	110s	7
[356]	Security Camera	Webp	3.8 kB	10 minutes	-
[357]	Experimentation purposes	CIRA	2.7 to 27 kb	30 s	1 (TTN compliance)
[358]	Experimentation purposes	JPEG	8kb	?	?
[361]	Weed and disease detection	JPEG	9kb to 28kb	49 to 165 seconds	5 to 17
[259]	Agricultural	JPEG	?	?	?

Table 3.4: Image transfer via LoRa literature survey

The research projects we've encountered lead us to several conclusions regarding heavy file transfers in LoRa networks. Firstly, while such transfers are possible, they are limited in terms of the number of transmissions per day. This means they may not be suitable for applications requiring frequent heavy data offloads. Based on our analysis, we believe that an application requiring more than 50 transfers a day would be impractical, and it would be advisable to minimize such transfers, thereby tailoring the application precisely to its needs. Secondly, heavy files must be compressed to reduce the transmission size. Ideally, the size of compressed files should be kept under 100 kb. Among the identified compression methods for images, three stood out: JPEG, JPEG2000, and WebP, with WebP appearing to be the most efficient. This emphasis on compression underscores the potential value of tinyML, as cloud-based ML algorithms would primarily have access to compressed data, implying a loss of information. Lastly, we want to highlight the inefficiency of the ALOHA protocol for large data file transmissions, as it could lead to network congestion when many packets are transmitted simultaneously. To mitigate this issue, various methods have been proposed, including CSMA protocol implementation, time slot allocation per node, and grouped packet acknowledgments

- **Firmware Update over the Air**

FUOTA is a critical requirement for IoT devices. The need for updates is inherent to any computer science project. This is due to multiple reasons: The upgrading of an application, the evolution of an application towards a different direction, calibrating a device in a specific environment, or implementing security updates to protect a device from newly discovered vulnerabilities. Specifically, in the case of IoT, these updates need to be performed over the air, i.e., wirelessly, as devices are often numerous and spread across vast geographical areas. This makes it challenging for operators to manually update each device by connecting to them through a wired connection. This is typically the case for LoRa agricultural networks. The authors in [362, 363] have provided comprehensive surveys on FUOTA for IoT. Additionally, the researchers in [364] have presented a survey detailing various security threats associated with firmware updates. There is usually two type of mode for FUOTA, push mode and pull mode. In "push mode," a centralized update server, whether on the cloud or locally, actively sends the latest firmware version to each device. This method's primary benefit is its immediacy, ensuring devices promptly receive updates when they're ready for them. However, a significant drawback is its high bandwidth consumption. As updates are sent individually to every device, this method is best suited for updating smaller groups of devices. On the

other hand, "pull mode" involves devices independently checking for available updates at regular intervals. If an update is found, the device will download it. This method is more bandwidth-efficient compared to push mode. Yet, it has its downsides. If the device checks infrequently, crucial updates could be delayed. Furthermore, if a device isn't in the right state to initiate an update, it might experience prolonged delays. Another consideration is that devices in pull mode don't necessarily register for these updates. While this might seem beneficial in terms of development simplicity, it's not ideal from device management and security standpoints. Both of these update methods can be implemented simultaneously within a single application, creating a hybrid approach.

In the specific case of LoRa and LoRaWAN, the FUOTA procedure has recently been standardized by the LoRa Alliance, a working group at Semtech, the company that owns the patented radio technology of LoRa. The detailed standard is available at [365, 366] and is illustrated in figure 3.10. It is recommended as a push-based method but could be adapted into a hybrid or pull method. The FUOTA process is managed by the Firmware Update Server, which collaborates closely with the Application Server that implements a firmware management layer. The File Distribution Server oversees the key parameters of LoRa FUOTA, which include multicast addressing, fragmentation, and clock synchronization. On the device side, the firmware must incorporate a boot-loader to oversee transitions between firmware versions. The various steps of the FUOTA process are as follows:

1. **Multicast addressing:** Typically, multiple devices need updates simultaneously. For efficiency, these devices can be grouped under a single address. Each device in a LoRaWAN network has a unique identity, comprised of a 32-bit device address and a set of AES₁₂₈ keys for unique identification and authentication. A device can also be part of up to four multicast groups. During multicast addressing, the application server identifies the devices to be updated, assigns them a specific address, and provides them with an application key for payload encryption and a session key for computing the message integrity check (MIC).
2. **Fragmentation:** Firmware images are typically large, often spanning several hundred kBytes, and cannot fit into a single downlink packet; thus, multiple packets are necessary. Given that LoRaWAN links can sometimes be unreliable, packet losses are common. This makes it difficult to identify which packets have been lost during multicast transmissions. To mitigate this, the fragmentation specification provides commands to ensure the reliable transfer

of large data blocks to individual or multiple devices. The application server sends information to the device about each transmission session, including the total number of packets it should receive, as well as the error recovery and fragmentation algorithms being used.

3. **Clock synchronisation:** Many LoRa devices operate in Class A mode to conserve energy. To simultaneously update multiple devices, the network server assigns a time slot to each device to be awake to start the transmission of the update file. Since device clocks can drift over time, the clock synchronization process ensures all devices maintain consistent timing by updating in tandem with the server. Upon reaching their designated time slot, devices switch to Class C mode to receive the update file.
4. **Transmission:** During transmission, the gateway sends a large number of packets to the device, which doesn't acknowledge receipt to prevent network congestion.
5. **Update** Upon receiving the firmware update, the bootloader checks its integrity before beginning the update process. If the received file has irreparable flaws, the update is discarded. After the update, the device reboots and usually returns to its default mode, typically Class A.

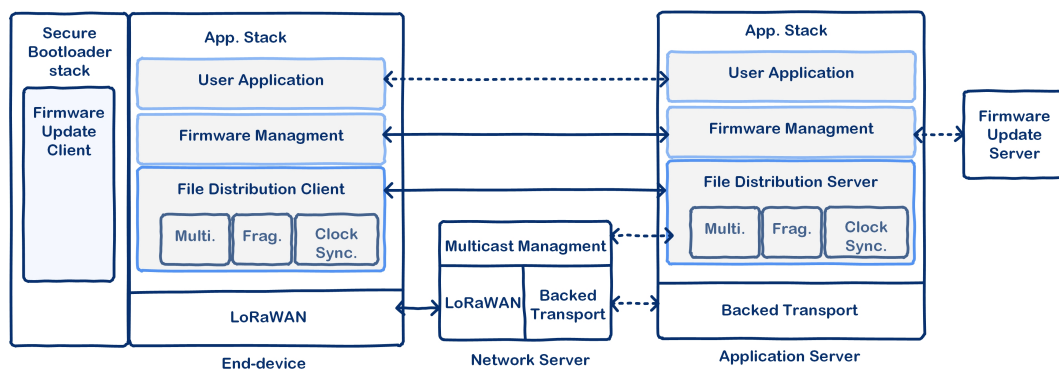


Figure 3.10: FUOTA architecture

Authors in [367] have studied the specificities of the standard in different network scenarios and have concluded for the efficiency of the method for firmware updates smaller than 100kb but only when transmitting with the best data rate available. Authors in [368] have shown that uploading a 128 kb firmware size took between 16 min and 8.36 hours depending on the data rate, this is acceptable timing in the best data rate scenario but it is way too long to respect duty cycle regulation. Authors in [369] have

proposed an efficient implementation of a FUOTA LoRa server with the open source LoRa server from the Chirp Stack. To the best of our understanding, no research has been done on updating TinyML models via LoRa. Nonetheless, we presume that the firmware update process for such applications mirrors that of more conventional ones.

Hardware

Hardware evaluations were carried out across multiple platforms. For devices running the TinyML algorithm, we selected several microcontrollers based on specific criteria:

- **Arduino's Portenta H7 [370] paired with the Portenta Vision Shield - LoRa [371]:** Arduino environment needs little introduction for those familiar with microcontrollers. Recently, Arduino shifted from its iconic 8-bit architecture to the more potent 32-bit system, facilitating advanced computations. The Portenta H7 represent this evolution with its Arm cortex microcontroller. It's complemented by a shield featuring the Himax HM-01Bo monochrome 320x320-pixel camera module and LoRa capabilities, powered by Semtech's SX1276 chip. The rationale behind using an Arduino is to make replicating our research easier, as the Arduino development environment is widely recognized and user-friendly for embedded system engineers.

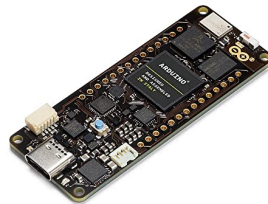


Figure 3.11: Arduino Portenta H7 and its LoRa Vision Shield

- **STM32 Nucleo pack LoRa [372]:** STM32 offers a pair of development boards designed for LoRa experimentation: a sensor node and a gateway. The sensor node utilizes the STM32 NUCLEO-Lo73RZ equipped with an SX1276 LoRa transceiver. Meanwhile, the gateway is built around the NUCLEO-F746ZG and incorporates the SX1301 gateway chip. In the realm of embedded devices, STM32 is regarded as a professional choice, and the STMCube ecosystem holds a prominent position in the industry. Furthermore, its 32-bit architecture enables the execution of complex algorithms. This made it compelling for us to evaluate its performance within our architecture. However, the kit did not include a cam-

era, so we integrated an OV2640 [373], a 2-megapixel RGB low-power camera

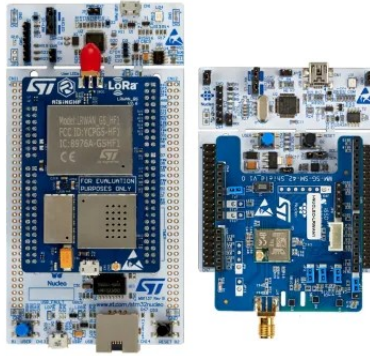


Figure 3.12: STM32 Nucleo pack LoRa

- **ESP32 CAM [374] and Hope RF Module [375]:** The ESP ecosystem is widely recognized as a cost-effective alternative to Arduino boards, primarily because it inherently supports Wi-Fi communication, facilitating rapid and affordable IoT device development. Recently, ESP has also begun transitioning to a more robust 32-bit architecture. They provide a development board that already includes an OV2640 camera [373] called ESP cam. To incorporate LoRa connectivity, we paired it with a low cost Hope RFM95W module, which features an SX1276 transceiver chip. The goal of this configuration is to showcase the potential to construct a basic TinyML camera node with equipment costing under \$10, thereby demonstrating the economic viability of such applications in agriculture.



Figure 3.13: ESP32 Cam and the attached Hope RFM95W module for LoRa connectivity

For the gateway, we used alternatively the one provided in the STM32 kit and a Laird Sentrius RG168. All communication devices were carefully selected to match European regulations and use the 868 MHz frequency.



Figure 3.14: Laird Sentrius RG168 Gateway

Finally for the software development and for running the simulation tools I used a HP elitebook Laptop computer with an i7 8th generation processor and 16 gigabits of RAM.

Software

Multiple software and tools were used during our research to develop our system and test it. Some of them in particular worth mentioning are:

- **STM32 Cube environment[376]:** This software was used for the programming of microcontroller. The STM32Cube development environment is a software platform offered by the company STMicroelectronics for their STM32 microcontroller series. It makes it easier for developers to create applications for STM32 microcontrollers.
- **VS code & PlatformIO[377]:** VS code is a famous IDE which has a rich ecosystem of extensions such as platformIO, a cross-platform, cross-architecture, multiple framework, tool for embedded systems development. It was used for the programming and interfacing of the LoRa network server and for the programming of ESP32 and Arduino Portenta H7 microcontroller.
- **The thing network[378]:** The thing network is an open source initiative developing a LoRaWAN server and sharing it freely worldwide. It was used during testing of the nodes behavior.
- **The ChirpStack[379]:** It is another open source LoRaWAN server software. It was used to handle the FUOTA process and the image offloading process.
- **FreeRTOS [380]:** FreeRTOS is an open-source real-time operating system (RTOS) for embedded devices. It offers multitasking support, allowing for the simultaneous execution of multiple tasks, and is optimized for low memory usage and strong performance. It was employed on the ESP32 nodes to facilitate firmware updates.

- **ArmMBDED OS [381]:** It is an open-source operating system for IoT devices. Designed for Arm Cortex-M microcontrollers, it simplifies the creation and deployment of code for nodes. It was used on the STM32 and Arduino Portenta.
- **Edge Impulse [382]:** Creating TinyML algorithms is a complex task involving multiple software such as TensorFlow Lite. To help with the rapid implementation of Embedded AI devices, Google offers a free-to-use platform for research called Edge Impulse. This cloud platform for Machine Learning operations (MLOps) [empty citation] performs training operations and data collection seamlessly through a graphical user interface. We used it to develop our TinyML model for counting strawberries.
- **Omnet++ and FloRa [383, 384]:** Many network simulators exist for LoRa, as indicated by the author in [385], but their conclusion is that their are equivalent and depend on the studied application. We chose OMNeT++ because it is an open-source, well-established C++ simulation framework primarily designed for creating network simulators. Various model frameworks, developed as individual projects, offer specialized functionalities, such as tools for sensor networks, wireless ad-hoc configurations, Internet protocols, performance evaluations, and more. In our case, one such independent framework is FLoRa [384], a LoRa simulator. FLoRa enables the establishment of LoRa networks with modules for LoRa nodes, gateways, and a network server. Application logic can be introduced as separate modules that connect with the network server. Both the network server and nodes support the dynamic adjustment of configuration parameters through Adaptive Data Rate (ADR). Moreover, energy consumption statistics are recorded for each node.

A recapitulatif table of Software and hardware solution is presented in table 3.5

	Network Communication	Embedded AI
Hardware	Laird RG168 gateway, Hope RFM95, STM32	Arduino Portenta H7, STM32, ESP32 CAM
Software	The thing network, The ChirpStack, Omnet++ and FloRa	Edge Impulse

Table 3.5: Hardware and Software Components for System Evaluation

3.2.3 . Phases

In this section, we describe the end-to-end phases implemented in our proposed architecture. Initially, we discuss the use-case scenario in normal mode to demonstrate its feasibility and energy efficiency. Subsequently, the same methodology is applied for the process of model re-training under degraded conditions that render the confidence score of the device node inefficient. The overall process is illustrated in Figure 3.15, with the steps for the normal behavior mode highlighted in blue and the degraded conditions mode highlighted in red.

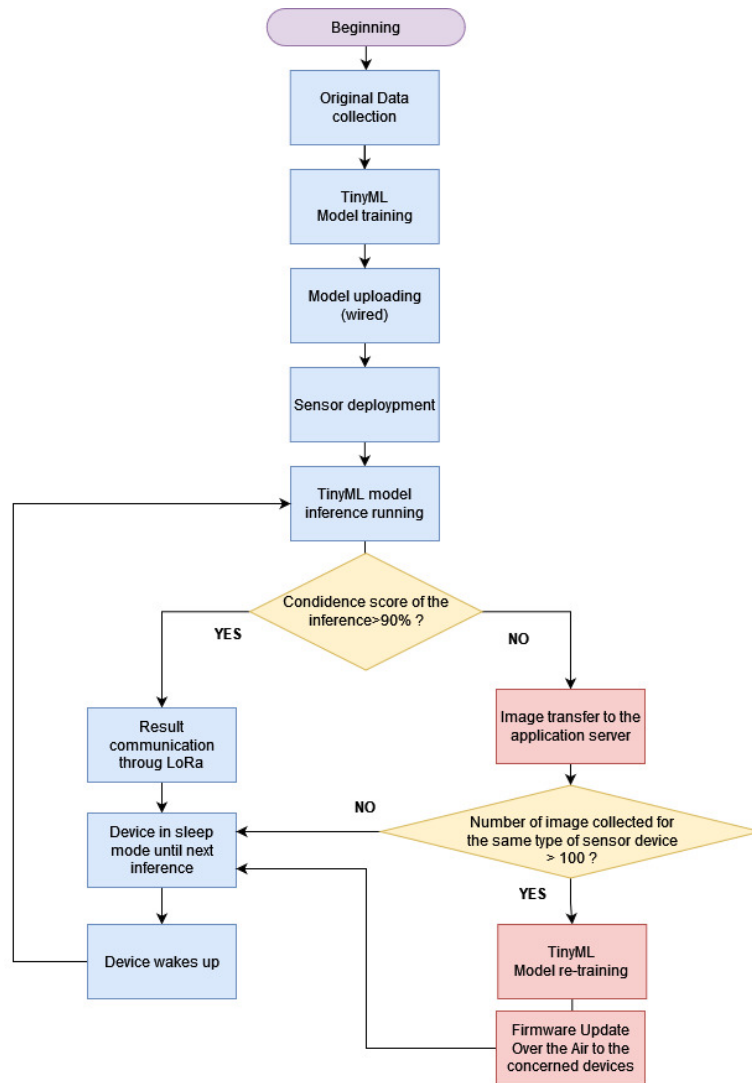


Figure 3.15: Overall process flowchart

Normal behavior mode

In normal behavior, the phases are as followed:

1. **Data collection:** In order to train the initial iteration of our model, we first need to collect data. Data can be obtained from a dataset; for example, in our application scenario, we started by utilizing the excellent fruit dataset from Kaggle [386]. However, the recommended approach in embedded machine learning is to collect data directly from the devices, since embedded camera modules often exhibit lower quality. Thus, in our scenario, the first dataset was collected through the camera device. We utilized the Edge Impulse tool to directly gather pictures from the Arduino Portenta and the ESP32 cam (which has the same camera module as the STM32). Once the data have been collected, they need to be labeled. In Figure 3.16, you can see image data labeled and collected by both the Arduino Portenta and the ESP32 cam, featuring various numbers of fruits.



Figure 3.16: Image data acquired through the Arduino Portenta are on the right, and through the ESP32 cam are on the left

2. **TinyML model training:** We trained our model to count the number of fruits in a picture using the FOMO (Faster Object, More Object) algorithm developed by Edge Impulse [387]. FOMO is a novel machine-learning algorithm that enables object detection on highly constrained devices. It allows devices to count objects, identify the location of objects in an image, and track multiple objects in real time, utilizing up to 30 times less processing power and memory than other similar dedicated algorithms, such as MobileNet SSD or YOLOv5 [388], which are two algorithms also available for constrained devices. The principle of FOMO is based on heatmap determination, wherein a picture is split into multiple smaller ones, and on each is performed a binary classification to determine the presence or absence of the object we seek to

count. The splitting factor of the original picture influences the processing time and power needed but allows better resolution for detecting objects that may be small or situated close to each other. Once the model is trained, it is tested over multiple data sets to determine its accuracy. In machine learning, accuracy refers to the fraction of predictions that a model gets right. It is calculated by dividing the number of correct predictions by the total number of predictions on a specified dataset. In our application, which is not critical, we set the minimum accuracy to 90%.

3. **Model uploading:** After the initial model is trained, it is uploaded to the device via a wired connection. The overall process of data collection, model training and deploying according to Edge Impulse is depicted in figure 3.17.

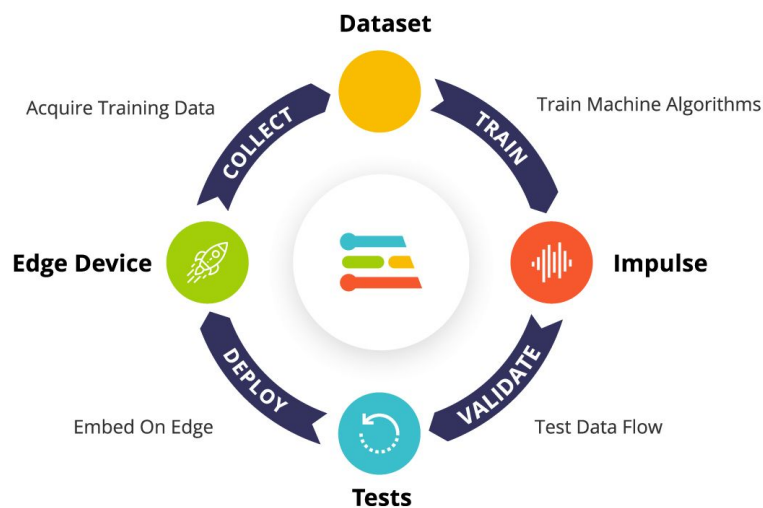


Figure 3.17: TinyML model creation process with edge impulse

4. **On-device inferring:** As soon as it is deployed, the device will begin making predictions. For our application, which involves counting strawberry fruits, we set the number of inferences performed to one per day, although the frequency can be adjusted according to the application's needs.
5. **Result transmission:** Once the result is obtained, it is transferred to the application server via the LoRaWAN network. The message consists of a single packet, with only the number of fruits included in the payload as an integer. After transmission, the device waits for an acknowledgment of reception from the gateway. If it does not receive it, it retries sending the message.

6. **Result interpretation:** The application server can interpret the data. Even if it only received a number, extra information such as the localization, the number of recent passages or the historical data values of the sensors can be retrieved from the knowledge implemented with the system described in chapter 4.
7. **Sleep procedure:** Once the device finished transmission it goes back in sleep mode. Sleep mode in microcontrollers refers to a low-power state that conserves energy when the device is inactive or when full power is not necessary. In sleep mode, certain functions or sections of the microcontroller are turned off or operated in a reduced-power state to minimize energy consumption, thereby extending the battery life of battery-operated devices. The exit from sleep mode can occur due to a scheduled procedure that is induced at the end of a timer, or due to an external signal. In our case, it is triggered by a timer.

Degraded condition mode

In degraded condition mode the phases are as follow:

1. **On-device inferring:** Every time the device runs the tinyML algorithms, it assigns a confidence score to the prediction. In machine learning, and specifically in the context of convolutional neural networks (CNNs) used for classification tasks, a confidence score is a value that represents the model's certainty regarding its prediction. After processing input data, the model generates a prediction; in our case, the estimated number of fruits. The confidence score (Cs), usually a value between 0 and 1, indicates how certain the model is that its prediction is correct, with a score closer to 1 indicating higher confidence. For example, in our specific case, the prediction might output a number of 3 strawberries with a confidence score of 0.95, meaning that the device is highly certain. However, if it counts 4 strawberries with a confidence score of 0.45, this indicates that this result should not be taken into account. In the case of successive inference processes that have low confidence scores, this might indicate that the model is not adapted to the local environment where the device is deployed and needs a new, better-adapted model. In situations where multiple predictions in a row (number to be determined by the application, in our case we set it to 4) have low confidence scores, the system enters degraded condition mode.
2. **Image data transmission:** To address the issue of local drift, the system needs to train another model; therefore, the application server requires new data. As explained in Section 3.2.2, transmitting images over

LoRa is a complex process. In our case, we propose the following algorithm:

Algorithm 1 Image transmission algorithm

```
if  $C_s \leq 0.9$  then
  Compress image to Webp format
  Convert image in Webp format to hexadecimal string
  Determine data rate for transmission
  Split the file into a number of packet  $N$  according to data rate
  Group packet together by a number of 10
  Add an application specific CRC code for each group.
  Establish connection with gateway
  while  $PacketSent \leq N$  do
    Send packet by group
    Server check the integrity of the received data
    if CRC does not match then
      Server asks for group Re-transmission
      Device re-transmit the group
    else
      Servers sends acknowledgment of reception
    end if
  end while
  Final error verification check
  Server acknowledge the reception
end if
```

The procedure begins by compressing the image to the WebP format, which is identified as more efficient than JPEG or JPEG2000 in existing literature [354]. The file is then prepared for transfer by converting it to a hexadecimal string. Once the data rate is selected, we determine the maximum payload size per packet and subsequently split our file accordingly. We group the packets in sets of ten to minimize network congestion, as each packet typically receives an acknowledgement in LoRa. To prevent any loss of information during this grouping, we add an extra Cyclic Redundancy Code to each group, allowing the application server to verify the successful receipt of each one. The transmission process then commences, and each group of packets is acknowledged by the server. If a group is not received, it is resent. Once all the groups have been retrieved, the server conducts a final verification of the overall received file and acknowledges the device.

3. **Server model re-training:** Once a certain amount of new image data has been collected, the server can re-train a new model. In our appli-

cation, we obtained a model with sufficient accuracy using 100 images. Depending on the specific task a smart sensor must perform, this number may be lower or higher. Images can be collected from one device or multiple devices monitoring the same phenomenon under the same conditions. Linking devices with similar characteristics can be challenging due to variations in hardware, applications, and the environments in which they are deployed, among other factors. Chapter 4 of this thesis will focus on semantic methods to link sensors together. The model retraining is quite straightforward and occurs in the same manner as in the normal behavior mode. It's important to note that in our experimentation, image data are labeled manually, requiring significant work, which could be a limitation of this solution. However, we could presume that more generic and precise AI models running on more powerful servers could automatically label the data collected by smaller devices with lower computational power. This machine-learning-oriented subject should be a topic for further research, posing the question: "How can we automatically label data collected at the edge for TinyML model re-training?"

4. **Firmware Update Over the Air:** After obtaining the new model, the final step in the process is to update the concerned devices. Once more, the server performs device grouping through semantic capabilities. When the server is prepared to update, it initiates the FUOTA procedure, as elucidated in 3.2.2. This procedure involves multicast addressing, file fragmentation, and clock synchronization preceding the transmission, followed by a reboot and update after an integrity check has occurred. This process is illustrated in figure 3.18, where the mode change from class A to class C during transmission, and back to class A at the conclusion. It is important to note that to avoid network congestion, there is no acknowledgment for individual packets during transmission. If the total packets received — and their integrity — do not pass the implemented error mechanism procedure, the device discards the FUOTA process and informs the server that a new update will be needed.

3.3 . Experimentation and results

3.3.1 . Embedded AI model

Model training

For our experiment, we trained our model in accordance with the previously described phases to identify strawberries, utilizing a collection of 100 pictures, each containing 0 to 10 instances of the fruit. We trained two distinct models: one using images from the Arduino Portenta Shield and its HM-01Bo monochrome

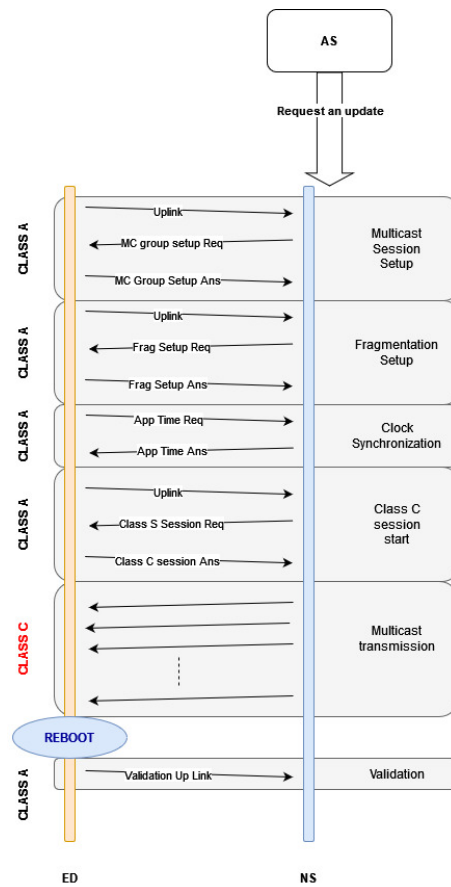


Figure 3.18: FUOTA Session

camera, and another using images from the ESP32 and its integrated OV2640 camera. Since the STM32 also utilizes the OV2640 camera, we will employ the same TinyML model for it. The Arduino’s camera, being black and white, produces images of lower resolution but of smaller size, which can be advantageous for image offloading. The type of images does not appear to influence the model size, as both library files retrieved from Edge Impulse after training are approximately the same size (242 kB for Arduino and 253 kB for ESP).

Regarding model performance, the model for the Arduino achieved a mean accuracy of **90.2%**, while the model for the ESP32/STM32 achieved a mean accuracy of **92.3%**.

Both libraries can be accessed from our GitHub [389], and implementation on platforms can be effortlessly tested using the example library provided by Edge Impulse [387].

Hardware deployment results

After training both TinyML models, they are ported to the various devices and tested multiple times to verify the system's robust behavior and obtain key metrics. We constructed a small wooden structure to hold the sensors and tested our architecture with varying numbers of strawberries. This test bench is shown in Figure 3.19.

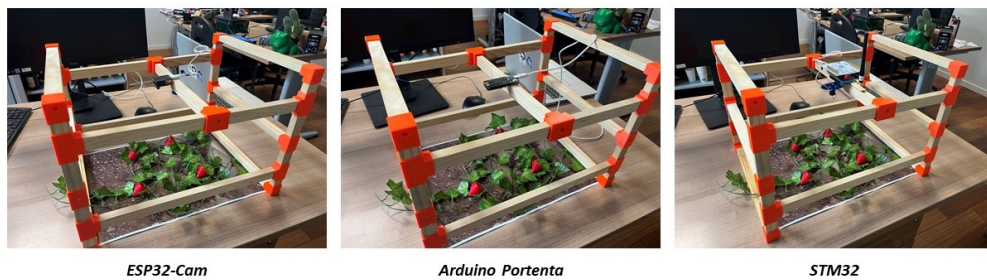


Figure 3.19: Test Bench Featuring the ESP32-Cam on the left, the Arduino Portenta in the middle, and the STM32 on the right

The technical characteristics of each device, with respect to RAM and flash memory capacity, are presented in Table 3.6.

	ESP32	Arduino Portenta	STM32
RAM	512 kb	1 Mb	100 kb
Operating frequency	8 MHz	48 MHz	32 MHz
Flash memory	448 Kb	2 Mb	1 Mb

Table 3.6: Hardware Characteristics

During a testing period consisting of 20 cycles for each device, we obtained a variety of results:

- **Arduino Portenta:** The results indicate that the estimated peak RAM usage is 243.9 kb and the firmware size is 77.5 kb, constituting 24.39% and 3.88% of Arduino's capacity, respectively, and leaving room for potential improvements. Over 20 inferences, the system's average inference time is 148 ms, and it predicted correctly for 19 measurements, signifying that the model's accuracy level is upheld.
- **ESP32-Cam:** The results show that the estimated peak RAM usage is 455 kB and the firmware size is 82.5 kB, which are 88.87% and 18.42% of the ESP's capacity, respectively. RAM usage is the limiting factor here, as a more complex model could exceed the RAM capacity for such a

device. Over 20 inferences, the system’s average inference time is 312 ms, and it predicted correctly for 17 measurements (85%), signifying that the training accuracy level of the model is maintained.

- **STM32:** For the STM32, the hardware characteristics are more constrained to achieve a trade-off with energy consumption. The firmware size, once compiled, is 93.2 kB, while the peak RAM usage is at 100 kB, representing respectively 9.32% and 100% of the STM32’s capacity. This constrained RAM usage seems to influence the inference time, as over our test of 20 predictions, the average time for an inference was 676 ms. With 18 out of 20 predictions being correct, the accuracy is again upheld, as field testing yields a 90% accuracy level.

We can observe from this hardware deployment testing that the models are performing as expected, confirming the efficiency of TinyML in agricultural scenarios. However, we also showcase the importance of hardware selection when developing an embedded AI application. In this specific scenario, some hardware, like the Portenta, appears somewhat oversized as most of its resources are unused, while, conversely, the STM32 is underperforming due to RAM capacity. On the other hand, the ESP32 seems to be the appropriate size. We have gathered the results in Table 3.7.

	ESP32	Arduino Portenta	STM32
Firmware memory usage	18.42%	3.88%	9.32%
RAM usage	88.87%	24.39%	100%
Inference time	312 ms	148 ms	676 ms
Accuracy respected?	Yes	Yes	Yes

Table 3.7: Hardware deployment results

Transmission

As soon as the TinyML model finished running, the number of fruits inferred is transmitted to the server through LoRaWAN with an integer encoded on one byte, meaning the payload size of the packet is also 1 byte. This also signifies that the message will be transferred in the form of a single packet. In our experimental setup, the Gateway and the node are located in the same room, approximately 10m apart, with a clear line of sight. The coding rate value is set at 1. We collected the average time of the transmission process over 10 transmissions for the typical data rates (DR0 to DR6) and devices. Additionally, we calculated the maximum number of transmissions per day according to the duty cycle. The results are presented in Table 3.8.

	DR0	DR1	DR2	DR3	DR4	DR5	DR6
ESP32	912.4	458.1	241.6	117.2	61.2	28.8	15.2
Max inference per day	947	1886	3576	7372	14118	30000	56842
Arduino Portenta	854.7	421.3	222.3	115.7	54.5	29.2	13.1
Max inference per day	1011	2051	3887	7468	15853	29589	65954
STM32	887.5	451.9	252.2	122.8	57.6	29.6	14.7
Max Inference per day	974	1912	3426	7036	15000	29189	58776

Table 3.8: Transmission Time in Milliseconds of Inference Result for Various Data Rates and maximal number of inference a day according to Duty Cycle

We have observed that the transmission time appears to be more dependent on the data rate than on the device. This can be explained by the use of the same transmission chip hardware, the SX1276, in each microcontroller. Finally, we also conclude that, overall, the transmission duration is quite short, minimizing power consumption and adhering to duty cycle regulations (a maximum of 864s a day for transmission). This allows devices to communicate their results between **947** and **65954** times a day.

Energy Consumption: Experimental Results and Simulations

The average energy consumption of each device in various operational states is provided in the manufacturers' datasheets and is presented in Table 3.9. We chose to use the theoretical values for energy consumption, as experimental readings yielded similar average results.

Mode	ESP32	Arduino Portenta	STM32
Sleep	10 μA	2.95 μA	0.29 μA
Run	190 mA	121 mA	105 mA
Transmission LoRa	211.5 mA	142.5 mA	126.5 mA

Table 3.9: Power Consumption for Each Device Across Various States

We can calculate the daily average energy consumption for each device by multiplying the time spent in each mode (specifically, 'run' during the inference process, 'transmission' during the transmission process, and 'sleep' during the sleep process) by the respective energy consumption of that mode and adding them together. The formula is as follows:

$$E_{daily} = E_{run} \times T_{run} + E_{transmission} \times T_{transmission} + E_{sleep} \times T_{sleep}$$

Using this daily energy consumption data, we can calculate the average battery life of a device utilizing a 2000 mA battery, a standard battery size for IoT devices. These results are presented in Figure 3.20.

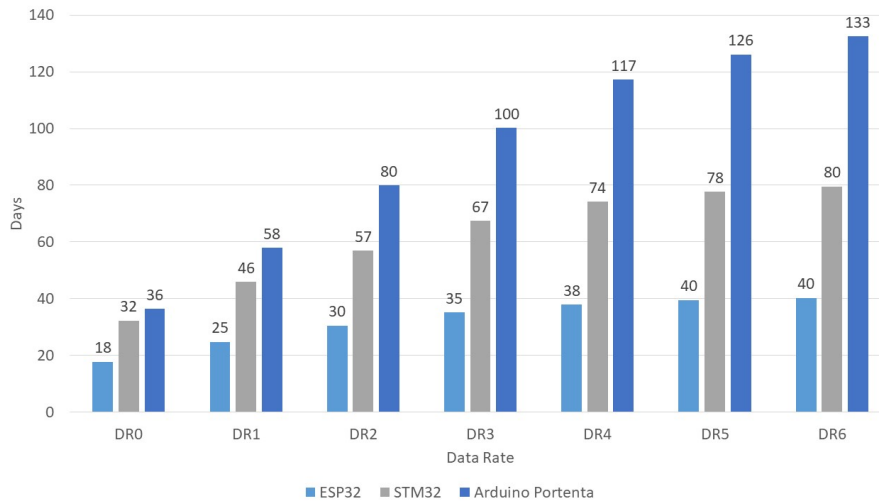


Figure 3.20: Battery Lifetime (in Days) for Various Data Rates, Based on Experimental Results and Theoretical Values

With our experimental system, we illustrate that a 2000 mA battery can sustain operation for between 18 and 40 days for the ESP32, 32 to 80 days for the STM32, and 36 to 133 days for the Arduino Portenta.

Our experimental results are somewhat limited since our test bench does not fully mirror the real-world usage of sensors deployed on a farm. To provide a more comprehensive overview of battery life, we input the previously collected values into the FloRa [384] simulator, which is based on Omnet++ [383]. This simulator models data collection over a one-hectare surface with 100 devices, all maintaining a clear line of sight. The input parameters provided to the simulator are compiled in Table 3.10. We evaluated the energy consumption of each device across various data rates over a year.

The results collected from the simulator provide deeper insight into the energy consumption of each device in a Wireless Sensor Network (WSN) deployed on a farm. The results are displayed in Figure 3.21. The simulation yields data regarding the overall energy consumption of each device for each data rate over a year of operation. The total amount of energy per device per data rate is presented in Table 3.11.

From the simulation results, we observed that for the Arduino and STM32, with data rates between DR2 and DR6, a battery of 10,000 mA would enable a device to easily last a year on a battery. The ESP32 would require more optimization to be able to run on a small IoT battery, but these results emphasize once again the need for precise hardware selection for every application.

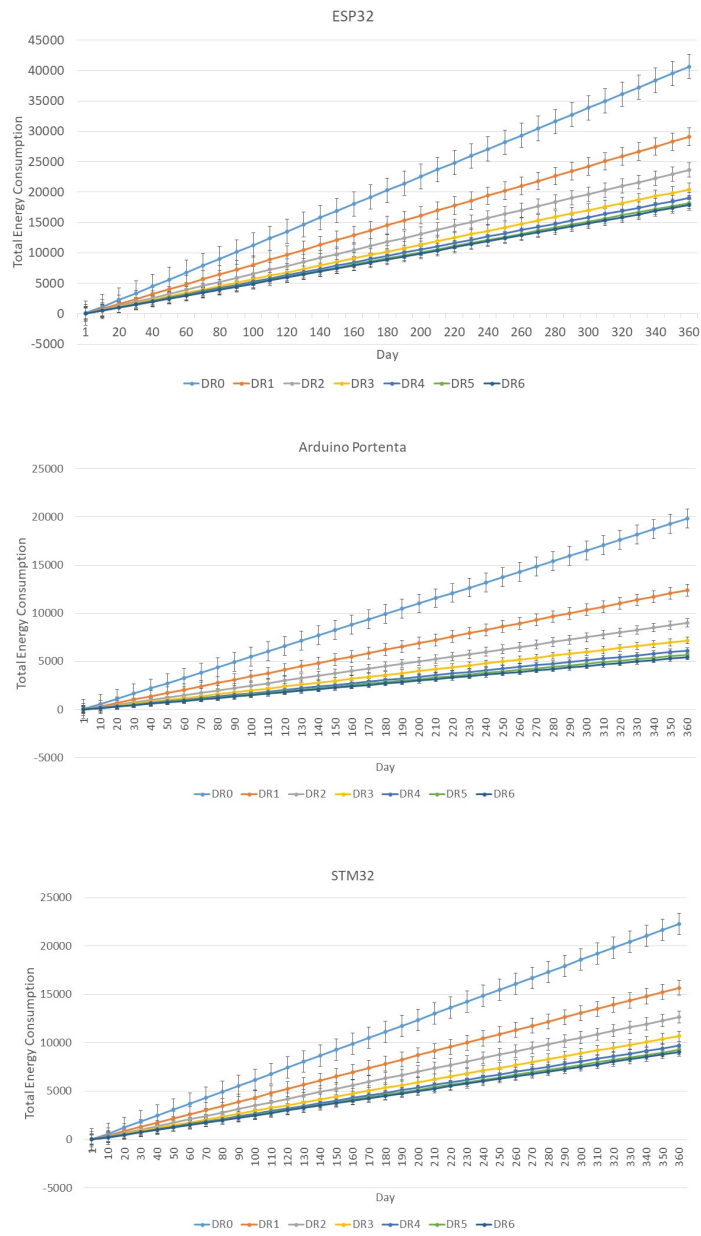


Figure 3.21: Simulation Outcomes for a One-Year Operational Period

Parameters	Value
Simulation Time	31556952s (1 year)
X,Y dimensions	100m x 100m (1 ha)
Number of Gateways	1
Number of Devices	100 (1 for 10 m^2)
Network topology	Star (1 year)
Carrier Frequency	868 MHz
Packet size	51,115 or 222 bytes <i>Dependent on the spreading factor (see 3.2.2)</i>
Spreading Factor	SF7 to SF12 <i>Dependent on the coding rate (see 3.2.2)</i>
Bandwidth	125 or 250 <i>Dependent on the coding rate (see 3.2.2)</i>
Coding rate	DR0 to DR6
Energy consumption run mode	<i>Depending on tested device (see previous values)</i>
Energy consumption transmission mode	<i>Depending on tested device (see previous values)</i>
Energy consumption sleep mode	<i>Depending on tested device (see previous values)</i>
Run time	<i>Depending on tested device (see previous values)</i>
Payload size	1 byte

Table 3.10: Simulation Parameters

From theoretical and simulation results, we conclude that our architecture is suitable for our scenario. We understand that by enhancing the battery size and performing energy optimization, each device could easily operate for an entire year or even longer. This proves the suitability of using TinyML and LoRa architecture for machine learning applications in agriculture, as farmers not only have access to AI at the edge but also only need to change the battery of their sensors infrequently. Furthermore, our experimentation could be further optimized in the future for different specific applications, making it even more energy-efficient. We also observe that energy consumption is largely dependent on the device, emphasizing once again the necessity for correct hardware selection when deploying an embedded AI model.

We also conducted additional analyses to showcase the advantages of our TinyML and LoRa architecture compared to the use of WiFi and cloud-based AI, utilizing only the Arduino Portenta for this purpose. Three scenarios were considered in our experiments, maintaining the same parameters as the previous experimentation—meaning one inference per day and a 2000 mA bat-

	DR0	DR1	DR2	DR3	DR4	DR5	DR6
ESP32	41203	29512	23941	20740	19299	18465	18115
Arduino	20098	12584	9134	7286	6225	5786	5507
STM32	22605	15901	12828	10836	9833	9402	9172

Table 3.11: Total Energy Requirement in mA for One Year of Device Operation

tery. In the first scenario, the TinyML algorithm operates directly on the microcontroller, and the inference result is communicated via LoRaWAN. The second scenario also involves the TinyML algorithm running directly on the microcontroller, but in this case, the result is communicated via WiFi. In the final scenario, only the captured picture is sent via WiFi to offload 10 kB of data, representing the transfer of an image and validating the impact of TinyML on energy consumption. After retrieving the average transmission time of a packet through WiFi and a 10 kB image through WiFi, we can calculate the energy consumption of scenarios 1, 2 and 3 by using the previously enounced formula. The results are presented in Figure 3.22.

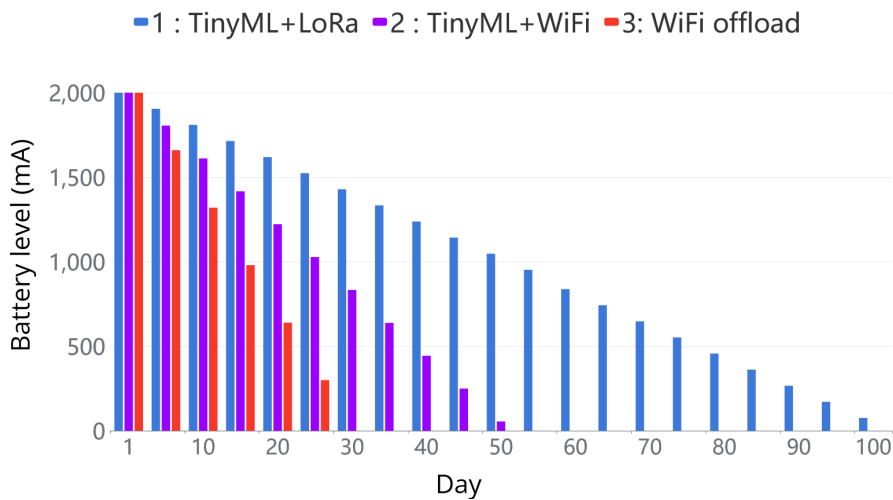


Figure 3.22: Evolution of Battery level over time

It appears that Scenario 1, where TinyML and LoRa are used, is the most energy-efficient one as it can last up to 105 days—three times longer than Scenario 3. Scenario 2 also shows that TinyML can save battery life in the case of full WiFi usage, as the battery lasts 1.5 times longer than in Scenario 3. The battery lifetime expectation results are compiled in Figure 3.23. These results validate our hypothesis that a system using LPWAN and TinyML is more energy-efficient than a system based on a more power-hungry communication method like WiFi or a cloud-based AI system.

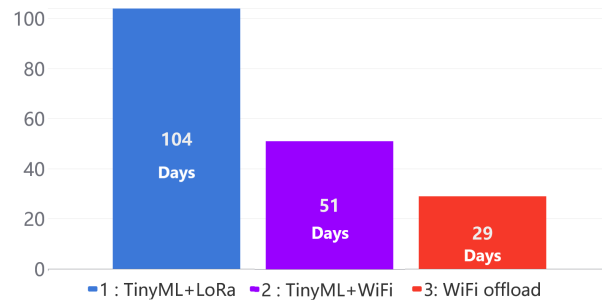


Figure 3.23: Battery Lifetime of the system in days

These results were presented at the *ICUFN: International Conference on Ubiquitous and Future Networks* in June 2022 [389], where they received encouraging feedback.

3.3.2 . Data offloading

Algorithm efficiency

After validating the suitability of our architecture in normal behavior mode, we addressed the degraded condition mode. In this section, we will focus on the image transmission process. We implemented the algorithm described in Section 3.2.3 into only the Arduino Portenta and ESP32 microcontrollers. We are concentrating on these two microcontrollers this time because the main complexity of our algorithm arises from the transmission side and the size of the image taken; whereas the ESP32 and STM32 utilize the same transmission chip and camera in their hardware configuration. The images taken from the Portenta and ESP32 can vary in size with each measurement. The average size of images taken by the Arduino Portenta and compressed with WebP over 10 images is 4.7 kb. For the ESP32, the average size of images, which are in color this time, is higher and reaches an average of 9.8 kb. The average transmission time for each image at different data rates is presented in Table 3.12. We maintained the same setup as previously, with only one node at a time, separated by 10m from the gateway, and with a clear line of sight.

From these results, we showcase the efficiency of our algorithm for sending images of different sizes, as it outperformed the other available methods presented in 3.2.2. We also obtained a result similar to the theoretical one regarding the duty cycle limitation.

Knowing that we only needed 100 pictures to train our TinyML model, we observe that the Arduino Portenta can retrain a new model in one day when communicating at data rates between DR3 and DR6. For the ESP32, retraining is possible at data rates between DR5 and DR6. However, devices, especially in agricultural scenarios such as ours, are usually not deployed in isolation, and multiple others are observing the same phenomenon. Therefore, data

Device	DR0	DR1	DR2	DR3	DR4	DR5	DR6
ESP32	117.60	66.82	30.00	16.70	9.41	5.37	2.67
Max image transfer per day	7	13	29	52	92	161	323
Arduino Portenta	56.40	32.05	14.39	8.01	4.51	2.58	1.28
Max image transfer per day	15	27	60	108	191	335	674

Table 3.12: Average Image Transmission Time (s) and Maximum Daily Transfers at Various Data Rates, According to Duty Cycle.

could be collected from multiple nodes, making it possible to transfer a sufficient amount of images for model retraining, even at the lowest data rate. For example, 10 devices equipped with the Arduino Portenta could each transfer 10 images at the lowest data rate, allowing a model to be retrained even in degraded network situations. Separating the data offload could also be beneficial for battery lifetime. Instead of having one device transfer 100 images, it would be more optimized to share this task among ten devices in order to maintain a balance in the battery level, avoiding one device running out of power more quickly than the others.

Energy consumption: Experimental result and simulation

We initially calculated the energy consumption of our system, basing our evaluations on our experimental results and the methodology previously proposed, during normal behavior mode. We determined the amount of energy needed to transfer 10, 50, or 100 images per device. These results are presented in Figure 3.24 for the ESP32 and in Figure 3.25 for the Arduino Portenta. Subsequently, we calculated the number of times the process of sending such quantities of images could be executed on a 2000 mA battery, and presented the results in Table 3.13.

The results conclusively affirm the energetic feasibility of our system. However, it is once again vital to note that, depending on the specific application, model drift may require more or less data offloading. Therefore, the system's sizing must account for this need.

On the other hand, simulation results were inconclusive. When inputting the same parameters as previously enunciated into the FloRa simulator, we noticed that the packet delivery ratio was only an average of 12.2%. This is attributable to the fact that our architecture did not implement any scheduling

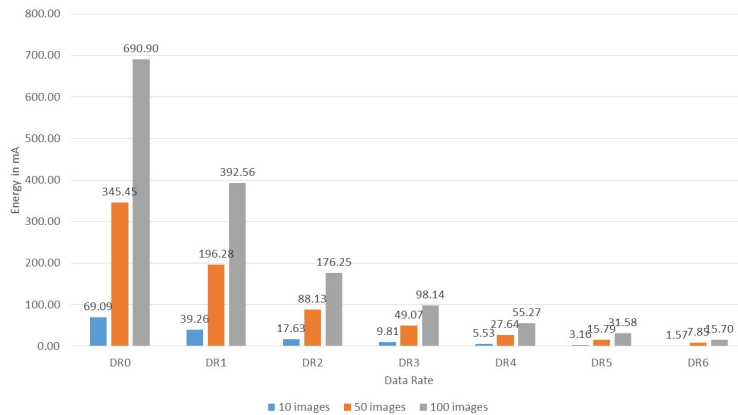


Figure 3.24: Energy consumption of the system for transferring various numbers of images using the ESP32.

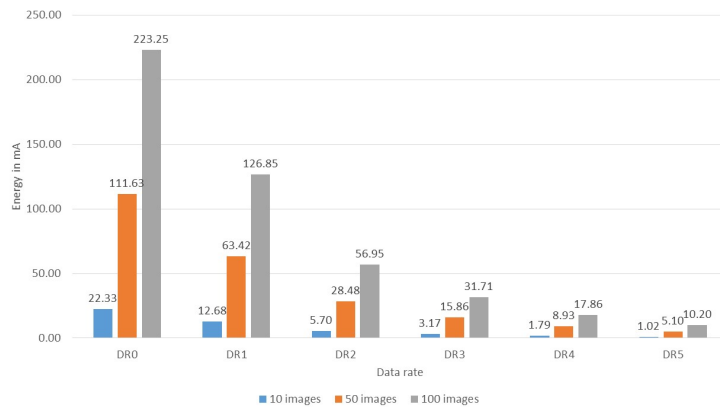


Figure 3.25: Energy consumption of the system for transferring various numbers of images using the Arduino Portenta.

methodology amongst the different nodes, meaning that every node tries to transmit their data simultaneously, leading to network congestion. Future work should, therefore, be conducted on scheduling.

Despite the fact that efficient scheduling could help ensure network reliability during image data offloading, we discussed the role 6G could play in such scenarios in a paper published and presented at the 6GNet conference in October 2023 [390]. Indeed, a device equipped with a multi-radio access system, such as LoRa for result communication and cellular for data offloading (and later firmware updates), could resolve the architecture issues identified. However, for this to transpire, 6G networks need to address the challenges of cellular technology for IoT devices, namely data plan pricing, low-power communication, and network coverage.

Device	Image Quantity	DR0	DR1	DR2	DR3	DR4	DR5	DR6
ESP32	10	29	51	113	204	362	633	1274
	50	6	10	23	41	72	127	255
	100	3	5	11	20	36	63	127
Arduino	10	90	158	351	631	1120	1960	3942
	50	18	32	70	126	224	392	788
	100	9	16	35	63	112	196	394

Table 3.13: Number of Image Offload Operations Possible with a 2000 mA Battery, According to Data Rate and Quantity of Images Sent in a Single Operation.

3.3.3 . Firmware update over the air FUOTA efficiency

In our final experimentation, we focused on the Firmware Update Over the Air Procedure (FUOTA). We implemented the protocol described in Section 3.2.3 using the open-source LoRaWAN server, ChirpStack [379]. We adapted the FUOTA server proposed by the authors in [391] to run on our Laird Gateway. The Arduino Portenta H7 operates Arm Mbed OS in charge of the reception of the update. First, we trained a new example model with new data and compiled it. The size of the updated model is 83.6 kb. The FUOTA process is long as the firmware size is substantial. We compiled the average time measured for 10 FUOTA process for a node device (reception and update) with the same firmware according to different Data Rates (DR) in Table 3.14.

	DR0	DR1	DR2	DR3	DR4	DR5	DR6
Arduino Portenta	552	475	412	302	245	176	153
Duty Cycle limitation	1	1	2	2	3	4	5

Table 3.14: FUOTA process duration in seconds (s)

From these initial results, we conclude that updating the TinyML model is feasible with LoRaWAN. As the FUOTA process is punctual, we can also assert that the duty cycle is respected, especially since it is the gateway that utilizes its duty cycle for transmission, the node device only sending acknowledgment messages. Therefore it is also of utmost importance for the gateway to correctly group the devices that need to receive the same update, as done in the multicast addressing procedure of the FUOTA recommendation. If multiple devices need different updates, the system should carefully schedule those FUOTA processes to avoid overflowing its duty cycle, especially with

lower Data Rates.

Energy consumption evaluation

To demonstrate energy consumption, we considered two scenarios, both employing a DR6 data rate. In each scenario, the device spends the majority of the time in standby mode and only wakes up once a day to perform a single TinyML inference and transmit a small telemetry message to the server. Additionally, the device wakes up once a week to be updated with new firmware of 100kb size. In the first scenario, all communications are performed with LoRa, while in scenario 2, WiFi is used as a point of comparison. For our experiment, we utilized the simple energy evaluation model from the INET framework of Omnet++ [383]. The simulator evaluates the lifetime expectancy of the battery and runs until both scenarios exhaust a 2000 mA battery. Results are presented in Figures 3.26 and 3.27. It appears that using LoRa in such an application could represent significant energy savings compared to WiFi usage.

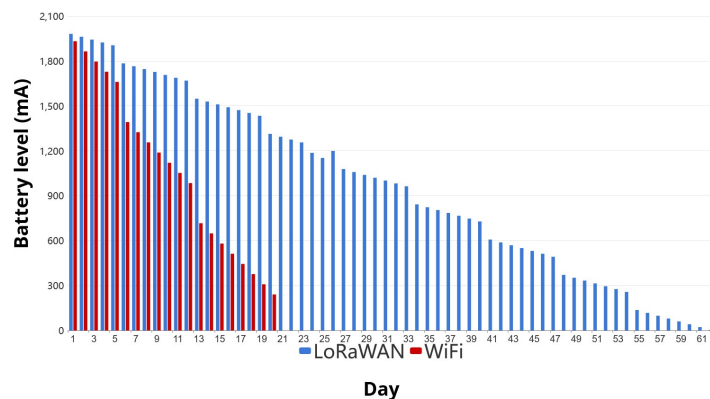


Figure 3.26: Evolution of Battery level over time

Furthermore, we propose an evaluation of the battery lifetime for various sizes of firmware updates for the same parameters as before (Arduino Portenta only and DR6). Results are presented in Figure 3.28 and Figure 9. As expected, the size of the update negatively impacts energy consumption.

The overall conclusion regarding the FUOTA process is that it is suitable for a TinyML model and energy-efficient. Once again, the frequency of the FUOTA process might differ depending on the application. Therefore, careful sizing of the battery should be undertaken to ensure good device behavior.

This work on FUOTA has been presented at the International Telecommunication and Network Application Conference (ITNAC) in 2022 and has received encouraging feedback [392].

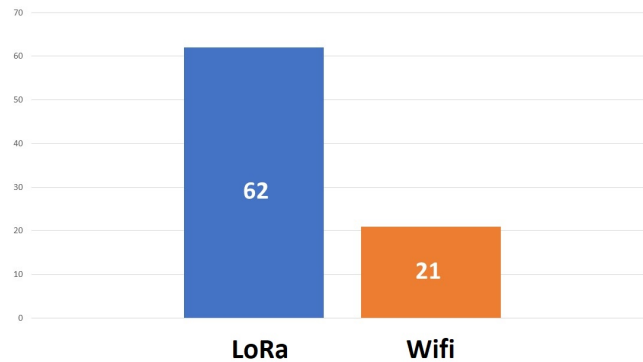


Figure 3.27: Battery lifetime of the system in days

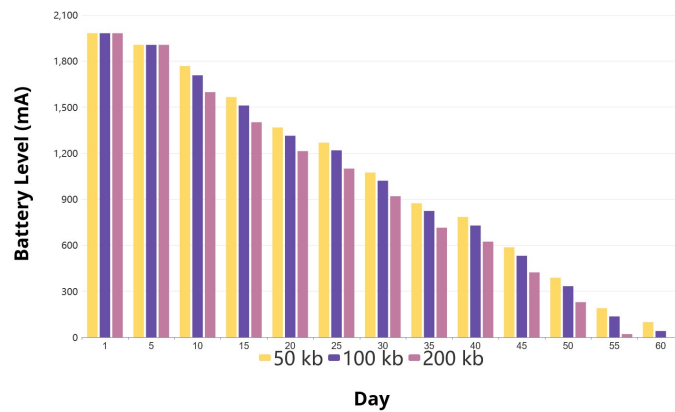


Figure 3.28: Evolution of Battery level over time for various size of firmware

3.4 . Conclusion

Promising results have emerged from this chapter, which focuses on energy-efficient embedded AI deployment in an agricultural context. Initially, TinyML is identified as a compatible technology for agricultural applications. Especially when paired with a Low Power Wide Area Network (LPWAN)—in this instance, LoRa—it can furnish significant energy savings. Such devices, by processing data at the edge, circumvent dependence on the cloud, thereby enhancing privacy and reducing latency. Furthermore, they avoid transmitting heavy data files like images or sounds, a process that is not only energy-intensive but also demands more robust networks, such as cellular or WiFi. These networks, however, bring their own challenges in coverage, range, energy consumption, and cost.

Utilizing LoRa and LoRaWAN enables devices to operate longer on battery power—a critical aspect for agricultural sensor networks—by transmit-

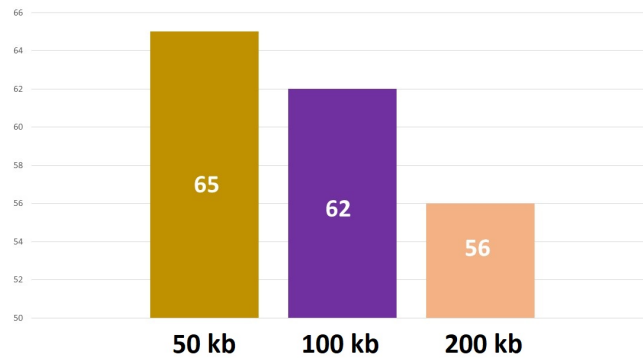


Figure 3.29: Battery lifetime of the system in days for various size of firmware

ting small messages over vast distances with minimal energy expenditure. LoRa, with its open-source nature, low cost, and the capability for private deployment on unlicensed bands, is particularly appealing for farmers. This is especially true in rural areas, where farms may lack adequate coverage from networks provided by professional carriers.

Nevertheless, TinyML implementation comes with its limitation, notably its sensitivity to drift. Once deployed in a specific environment, the embedded generic model will necessitate updates with local data. However, on-device learning for IoT devices is not feasible due to the computational limitations of the hardware. To address this issue, we introduce an architecture in this thesis that encompasses sporadic data offloading, server model retraining, and a Firmware Update Over The Air (FUOTA) process for TinyML devices. We have demonstrated that retrieving heavy files punctually, such as images via the LoRa network, is not only feasible but also energy-efficient, as is the FUOTA procedure.

Future work should explore various aspects of our proposal. Firstly, a scheduling process should be implemented for image offloading to prevent network congestion when too many devices attempt to offload simultaneously. This issue also pertains to the FUOTA process, where the gateway must efficiently schedule offloading and group devices. On a wider scale, we ponder the efficacy of a multi-radio access technology for such applications. While transmitting inference results via LoRa is recommended for energy-saving purposes, utilizing alternative wireless networks for data offloading and FUOTA could safeguard optimal network performance, particularly when nodes are densely populated in an area. Unfortunately, we've demonstrated that other current wireless networks are also inadequate for our application type. Thus, we suggest that future work on 6th generation telecommunications should consider these specific IoT needs. Finally, it is noteworthy to men-

tion ongoing efforts by various companies, like Starlink, and states to deploy satellite constellations that provide internet access even in remote areas, presenting a potentially valuable opportunity for IoT in agriculture.

4 - A knowledge-based approach for IoT platform management in agroecological farm

4.1 . Introduction

In the previous chapter, we presented an energy-efficient architecture for implementing smart sensors in an agricultural wireless sensor network, focusing on the perception aspect of our IoT platform. Now, we shift our focus towards the decision-making part of the platform, enabling our system to make informed decisions based on the sensor network. These decisions may pertain to farm management actions or sensor network maintenance operations. As we demonstrated in Chapter two, an effective tool for managing a wide range of data and agents, as is the case in agricultural IoT, is the creation of an expert system capable of making decisions with the assistance of a knowledge base.

A knowledge base allows for a formal representation of a context by providing a systematic classification of elements, their properties, and the relationships that exist between them. By employing an additional tool known as a reasoner, it becomes possible to imbue the objects in the knowledge base with a layer of intelligence through logical rules. This set of rules can address various situations, such as identifying the current state of a sensor or determining the appropriate course of action in the event of sensor malfunction. This approach enables the system to possess the capacity to perceive and interpret the environment, which are critical requirements for applying Agroecology knowledge. As we discussed in the introductory chapter of this thesis, Agroecology is a form of agriculture heavily reliant on the local environment, the distinct biological characteristics of the individuals within it, the crops being cultivated, and other factors.

In this chapter, we will introduce our work related to the development of a knowledge base for Agroecological farm procedures and the management of Wireless Sensor Networks in the context of smart sensors. We will introduce the concept of ontologies and explain how to construct them. Subsequently, we will describe the various resources that constitute our own ontology, along with the rules implemented to manage them. Finally, we will demonstrate how the ontology is employed by a reasoning algorithm to facilitate informed decision-making within our system.

4.2 . Definition

4.2.1 . Ontology

An ontology is a formal, explicit representation of knowledge that defines the concepts, relationships, and properties within a specific domain [393]. It serves as a structured model or framework for organizing and capturing knowledge about a particular subject area. An ontology delineates both the concepts existing within a particular domain and the connections that exist among these concepts. Various ontology languages offer distinct capabilities. The latest advancement in standardized ontology languages is exemplified by OWL (Ontology Web Language), developed by the World Wide Web Consortium (W3C) [394]. OWL ontologies are an implementation of Description Logic (DL) [395] which is a decidable subset of First Order Logic. OWL exists in various forms with increasing levels of expressiveness, namely OWL Lite, OWL DL, and finally OWL Full. The ascending levels of expressiveness enable the representation of more extensive knowledge, but they also entail higher computational requirements. In this thesis, we opted for OWL Full since computational efficiency was not a primary concern. An OWL ontology consists of Classes, Properties, and Individuals:

- **Classes:** A class denotes a fundamental concept employed for the categorization and representation of entities or objects possessing shared characteristics, properties, or attributes within a particular domain or knowledge domain. For instance, the "plant" class encompasses all individuals that fall under the category of plants. Classes are organized hierarchically, with superclasses representing more general categories and specialized sub-classes representing more specific ones. In the context of OWL, class hierarchies are not limited to tree-like structures; they allow for the use of multiple inheritance. As an example, the "tomatoes" class can inherit from both the "plant" and "fruit" classes, signifying that all tomatoes are considered both plants and fruits, and consequently, they possess the corresponding attributes. Classes can be defined using various logical expressions, including enumeration of possible instances for the class, forming unions with other classes, creating intersections between two classes, expressing negation (indicating a class's opposition to another), defining disjunction (stating that a class cannot belong to two classes simultaneously), and other methods outlined in [393].
- **Properties:** Properties serve as binary relations that establish connections between individuals of specific classes, facilitating relationships between them or linking an individual to a literal value. For example, the "hasLocation" property can be employed to connect an individual

crop with an individual field, denoting the spatial association between them. In the same context, a field of the same type could possess a property called "hasGPScoordinate," pointing to a literal value. Properties can be defined in various ways, including symmetric properties, which imply that if a relation exists between class A and class B, it also holds between class B and class A. Similarly, inverses for concurrent relations indicate that if one individual is the child of class A, then class A is the parent of that child. Various types of properties are available in the OWL documentation to cater to different modeling needs.

- **Individuals:** also known as instances or objects, refer to specific, concrete entities or members of a class within a domain. Unlike classes, which represent general categories or types of entities, individuals represent actual, distinct objects or data points that exist within the real world or a modeled domain. Individuals can be thought of as the tangible instances or examples that belong to a class in the ontology. As an illustration, the "cow" class may encompass various specific individuals, such as Marguerite, Anatole, and Luciole. Additionally, these individuals can also be considered instances of other classes; for instance, Marguerite could be an instance of both "Dairy" and cow; "Anatole" could also belong to the "Bull" class, each representing distinct categorizations.

Ontologies can also be represented as a Knowledge Graph (KG). An example of an KG for an ontology related to Points of Interest (POI) is depicted in Figure 4.1. In this illustration, various classes, instances, and properties are employed to model knowledge pertaining to a POI. Notably, the illustration demonstrates that individuals representing the same entity in different classes can be connected as equivalents, as demonstrated with the "Wine region."

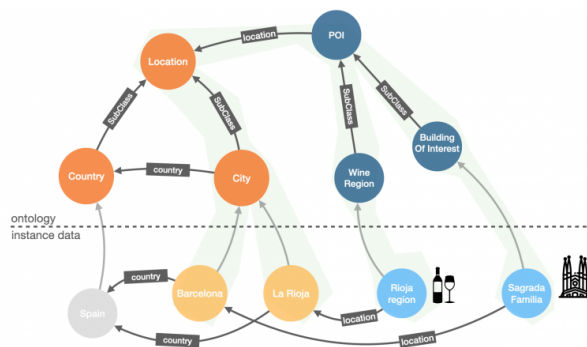


Figure 4.1: Ontology and instanced data comparison

4.2.2 . Ontology development with OWL and Protégé

Designing an ontology entails several important considerations. Firstly, it's crucial to recognize that there are often multiple valid approaches to modeling a domain, with the most suitable choice depending on the specific application and potential future extensions. Additionally, ontology development is an iterative process, requiring refinement and adaptation as one's understanding of the domain deepens or new requirements arise. Finally, when crafting an ontology, it's essential to ensure that the concepts within it closely reflect the objects and relationships found in the domain of interest, typically corresponding to nouns (representing objects) or verbs (representing relationships) as they appear in domain-related sentences. This alignment enhances the ontology's ability to effectively capture and represent real-world knowledge. In details, the steps to build an ontology are as followed [396]:

1. *Define the Scope and Purpose:* One should start by clearly outlining the ontology's scope and objectives. This involves specifying the domain it covers and the goals it aims to achieve. This initial step aids in identifying the concepts and relationships to be included.
2. *Identifying Concepts and Classes:* Next, identify and compile the essential concepts, entities, or classes pertinent to the domain. These serve as the foundational elements of the ontology. For example, in a medical ontology, concepts might encompass "disease," "symptom," "treatment," and "patient."
3. *Defining Properties and Relationships:* Determine the properties and relationships that connect these concepts within the ontology. Properties describe attributes or characteristics, while relationships specify how concepts relate to each other. For example, in a social network ontology, you might establish a "friend_of" relationship between individuals.
4. *Creating a Taxonomy or Hierarchy:* Organize the concepts into a hierarchical structure, often referred to as a taxonomy or class hierarchy. This hierarchy illustrates the subsumption relationships between classes. For instance, in a biological ontology, "Mammal" may be a subclass of "Animal."
5. *Specifying Constraints and Rules:* Define constraints, rules, and axioms to express the logical relationships and limitations within the ontology. This ensures semantic consistency and coherence. For instance, you might stipulate that an individual can belong to only one "species" class.
6. *Populating the Ontology:* Populate the ontology with instances or individuals that represent real-world objects or data within the domain. For

instance, in a geographical ontology, you may create instances for specific countries, cities, and landmarks.

To build our ontology we will be using the Protégé software proposed by Stanford University [397]. Protégé is a widely used and highly regarded open-source software tool for ontology development and knowledge modeling. It has been a staple in the field of ontology engineering for many years. Protégé provides a user-friendly environment for creating, editing, and managing ontologies, allowing users, including ontology engineers and domain experts, to define concepts, relationships, and constraints within a domain. It supports various formal ontology languages, including OWL (Web Ontology Language), making it versatile for different ontology projects. Protégé also offers a range of plugins and extensions, enabling customization and integration with other tools and services, making it a valuable resource in the development of knowledge-based systems and the Semantic Web.

4.2.3 . Reasoning

In ontology engineering, a reasoner is a software tool or component that plays a crucial role in semantic reasoning and inference. It is designed to analyze and draw logical conclusions from the information and relationships encoded within an ontology. Reasoners use the axioms, rules, and constraints specified in the ontology to perform tasks such as:

- *Consistency Checking:* Reasoners can determine if an ontology is logically consistent, which means there are no contradictory statements or errors within the ontology.
- *Classification:* They can classify individuals (instances) into appropriate classes based on the ontology's hierarchy and defined relationships.
- *Inference:* Reasoners can infer new information or relationships that are implied by the existing ontology. For example, they can deduce that if an individual belongs to a certain class and certain conditions are met, then it also belongs to another class or has specific attributes.
- *Query Answering:* Reasoners can answer complex queries about the ontology, helping users retrieve relevant information and make logical deductions.
- *Validation:* Reasoners can validate whether an ontology adheres to specified modeling patterns, constraints, and best practices.

One of the most well-known and widely used reasoners in the field of ontology engineering is the "Pellet" reasoner [398]. Pellet is an open-source reasoner that supports the OWL (Web Ontology Language) standard and is integrated with the Protégé ontology development environment. It provides ro-

bust reasoning capabilities for tasks such as consistency checking, classification, inference, and query answering. Pellet is recognized for its performance and scalability, making it a popular choice for ontological reasoning tasks in various applications. Therefore it will be the reasoner used in our work.

In more complex scenarios, a reasoner can be equipped with logical rules. A rule, in essence, is a philosophical concept wherein, assuming the truth or falsehood of a set of propositions, we can deduce the truth of another independent proposition. Rules consist of two distinct components: an antecedent set of conditions and a consequent set of actions. In computer systems, they are frequently depicted as statements in the format: "If <conditions> Then <actions>." There is multiple way to implement rules in knowledge base. In this work we will be using SWRL.

SWRL, which stands for Semantic Web Rule Language, is an ontology language extension designed to enable the representation of rules within ontologies in the Semantic Web context. SWRL allows ontology developers to express rules that define conditions and actions based on the ontology's classes and properties. A simple example use of SWRL rules would be to assert that the combination of the *hasParent* and *hasBrother* properties implies the *hasUncle* property. It could be written in SWRL this way:

$$hasParent(?x1, ?x2) \wedge hasBrother(?x2, ?x3) \Rightarrow hasUncle(?x1, ?x3)$$

Utilizing this expression, the reasoner can make the deduction that when X_1 is identified as the offspring of X_2 , it logically follows that X_3 , who serves as the sibling of X_2 , concurrently assumes the role of X_1 's uncle.

4.3 . Proposed model

For the knowledge base of our IoT platform related to agroecology, we have chosen to propose three main classes: WSN (Wireless Sensor Network), Crops, and Farm. The overall model is presented in Figure 4.2. Our approach involves implementing elements from the Semantic Sensor Network [316] for describing the WSN and drawing inspiration from the Plant Ontology available on Planteome [326] for describing plants. Additionally, we have taken cues from the work of authors in [329] to model pest and disease interactions. Furthermore, we will incorporate a rule layer to describe the interactions between these classes and generate valuable insights on how to manage farms and sensor networks.

As mentioned previously, we utilized the Protegé software to construct our ontology. We tested our ontology with O'FAIRe [399], an Ontology FAIRness evaluator, and obtained a fair score based on various criteria. The results are presented in Figure 4.3. The assessment of fairness for the ontology is based on various rules outlined by the authors [400]. The overall architec-

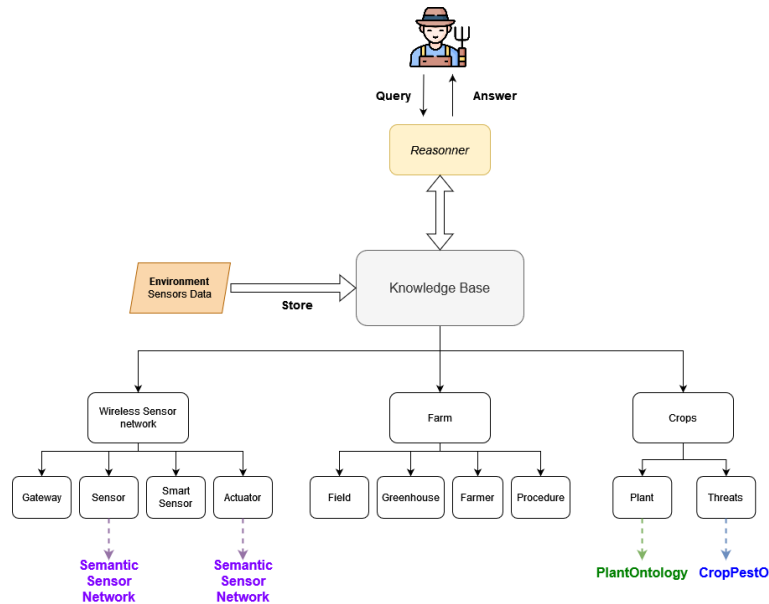


Figure 4.2: General Knowledge-Base model

ture of the constructed ontology is visible in Figure 4.4. In the next sections, we will describe in details each part of our ontology.

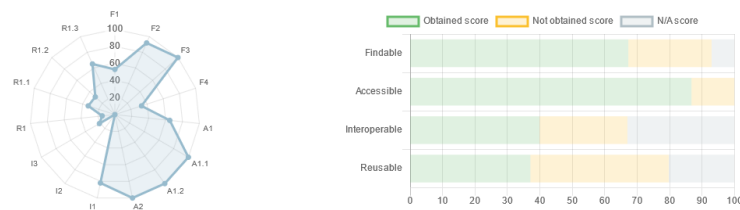


Figure 4.3: Ontology Fairness Evaluator Results

4.3.1 . Crops: Plant and Agroecology knowledge base

The crops class of our ontology is divided into two subclasses, namely 'plant' and 'threats.' The 'plant' class in our ontology is intended to represent the characteristics of each plant, while the 'threats' class groups together pests, diseases, and their corresponding countermeasures. The detailed graph of classes and properties is presented in Figure 4.6.

The details for each class are as follows:

- **Plant:** The 'Plant' class includes all the different types of crops that can be cultivated on a farm, such as wheat, tomatoes, salad, and more. While more complex information and knowledge are accessible for each



Figure 4.4: Ontology Structure in Protégé

plant through the linkage with the Plant Ontology (PO), we have defined some useful properties related to our application requirements:

- *isEquivalentTo* is an object property that links individual plants to equivalent classes in the Plant Ontology.
- *hasClimate* an object property that indicates the preferred types of climate for the plant.
- *hasLocation* is an object property that specifies the field or the greenhouse of the farm where the plant is situated.
- *hasMinSeedingTemperature* is a data property that indicates the minimum temperature required for the plant to commence development.
- *hasWaterRequirement* is a data property indicating the level of moisture required in the soil.
- *hasFavoriteTypeofSoil* is a object property indicating the preferred types of soils for the plant.
- *isSensitiveTo* is linking different threats to each plant. It is a symmetric object property with the property '*isMenaceTo*' of the '*Threat*' class, pointing to a disease or a pest.

On top of those basic elements, the novelty of our Knowledge base resides in the representation of what a plant needs from its ecosystem that could be provided by other plants and, conversely, the things it requires. The 'Agroecology_Bonus_Malus' class element is a subclass of the plant and models the things a plant can offer and could need from the ecosystem with the properties *canProvide* and *canRequire*. A quick example of the process is depicted in Figure 4.5, with the use case explained in the introductory chapter of this thesis being the Three Sisters companion planting (peas, corn, and squash), as depicted in Figure 1.10. We represent semantically the provided and required elements by each plant in its environment. By adding a rule in SWRL, the system is able to determine another property between two plants called *isGoodAssociation*, which is a symmetric property. This rule between two plants, *p1* and *p2*, on an agroecological element, *a1*, is in the form of:

$$canProvide(?p1, ?x3) \wedge canRequire(?p2, ?x3) \Rightarrow isGoodAssociation(?p1, ?p2)$$

- **Threats** The threat class is composed of three subclasses, namely the pest, disease, and countermeasure. The pest class encompasses the various types of biological attacks that can threaten crops. Similarly,

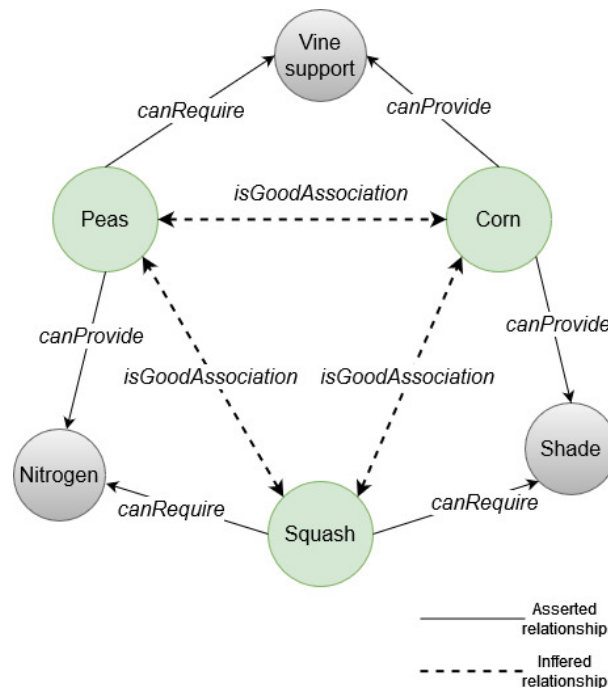


Figure 4.5: Relationship between different individuals of the Plant class and the inferred association based on the provided and required properties

the disease class includes various types of diseases. Each disease and pest is linked to an agroecological countermeasure by the property *hasCountermeasure*. This is a symmetric property with the property *protectAgainst*, which links every countermeasure to the pest or disease it aims to cure. Each threat and measure also has an *isMenaceTo* property, symmetric to each plant that it is a menace to.

4.3.2 . WSN: Sensors and actuators knowledge base

The goal of the WSN class is to represent the basic element of our sensor network, namely the normal Sensor, the Smart sensors and the Actuators. To represent the sensor characteristics, we inspired ourselves from the Semantic Sensor Network recommendation by the W3C [316]. The different elements are presented in figure 4.7 and are described as follows:

- **Sensor:** The sensor class represents classic telemetry sensors and groups various types of sensors. For the purpose of experimentation, we represented only a fraction of available agricultural sensors. These sensors are for measuring Air Humidity, Temperature, Soil Moisture, Sun exposition, wind speed, and rain level. Each sensor has different properties:
 - *hasValue* is a data property that represents the numerical value of the measurement.

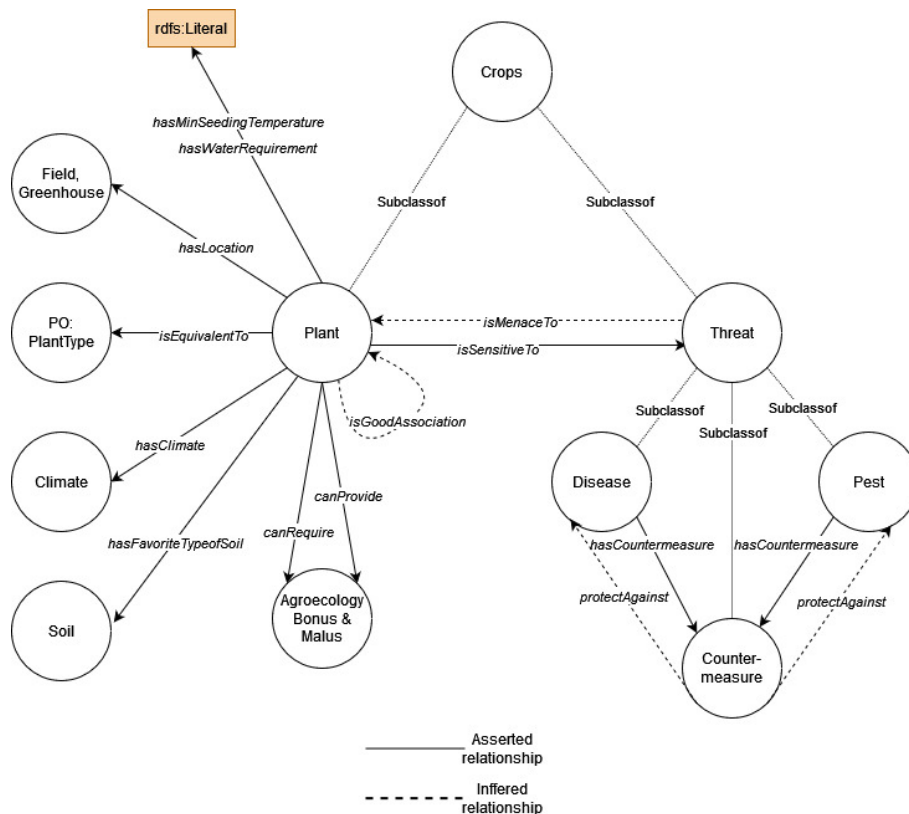


Figure 4.6: Ontological representation of the Crops class

- *hasLocation* is an object property that specifies the field or the greenhouse of the farm where the sensor is situated.
- *hasBatteryLevel* is a data property representing the battery level of each sensor.
- **Smart Sensor:** The smart sensor class is a bit more complex. First of all, they are linked to a firmware version through the property *hasFirmware*. The firmware class itself has a boolean data property of *isUpToDate* representing the state of the sensor. Another pertinent property, denoted as *performFUOTA*, serves to establish a connection between a smart sensor and the up-to-date Firmware version with whom to perform a FUOTA procedure. The smart sensors also have different types of applications. For the purpose of experimentation, we represented only three types of smart sensor applications, namely detecting pests, detecting diseases, and counting fruit. For the pest and disease subclasses, they are linked to the pests and diseases represented in the plant class by the object property *isDetecting*. Finally, all subclasses have the following properties as well:

- *hasValue* is a data property that represents the numerical value of the measurement. In case the smart sensor is counting fruit, it will have the value of the number counted; in case it is a sensor detecting a phenomenon, it will take the value 1 when the phenomenon is detected.
 - *hasLocation* is an object property that specifies the field or the greenhouse of the farm where the sensor is situated.
 - *hasBatteryLevel* is a data property representing the battery level of each sensor.
- **Actuator:** The actuator class represents the different machine-actionable systems in the farms. Again, a vast variety of them exists. In our system, we only represent the irrigation valve and the greenhouse fan system. We use the following properties to manage them:
 - *isActive* is a data property that represents the state of the actuator. Is it active or not.
 - *hasLocation* is an object property that specifies the field or the greenhouse of the farm where the actuator is situated.

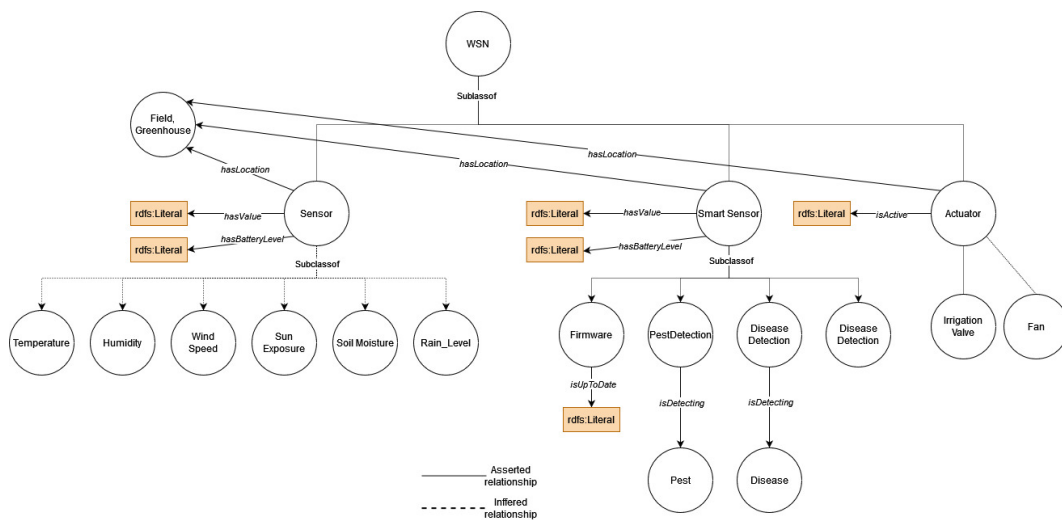


Figure 4.7: Ontological representation of the WSN class

4.3.3 . Farm

The farm class represents the various entities composing a farm and is presented in Figure: 4.8. Once again, there is a vast variety of entities that

could be implemented on the farm, but we focus on the farmer, the greenhouse, and the field. Additionally, we created a specific class for farmers regarding the procedures they have to implement. The farm itself has two properties, which are:

- *hasGpsCoordinate* is a data value representing the position of the farm in the world. This will allow our system to extract weather and climate data according to the farm's position.
- *hasCurrentWeather* is an object property pointing to a weather object that represents conditions such as Storm, Rain, Sun, Cloudy. Here, we could implement a Weather Ontology proposed by other research.

The different elements of the farm are the following:

- **Farmer:** This class represents the humans in charge of fields and greenhouses. This relation is represented with the property *isInChargeOf*, which links a farmer to one or multiple instances of the field and greenhouse class. This property is symmetric, with a *managedBy* property linking a field or a greenhouse to a farmer. The linkage between the farmer and each procedure is established through a property denoted as *isPerformingX*, where 'X' represents the particular type of procedure.
- **Field:** The field (similar to the greenhouse class) represents the spatial area where crops, sensors, and actuators are deployed. It is linked to the class Soil by the property *hasSoilType*.
- **Greenhouse:** The greenhouse class is similar to the field class but for greenhouses.
- **Procedure:** The procedure class represents the actions implementable by farmers. Here, too, the potential procedures are not exhaustive, and we focus on seeding procedures, harvesting, and applying countermeasures and sensor battery management. Once Again the linkage between the farmer and each procedure is established through a property denoted as *isPerformingX*, where 'X' represents the particular type of procedure (For example *isPerformingBattChange* for the procedure regarding changing the battery).

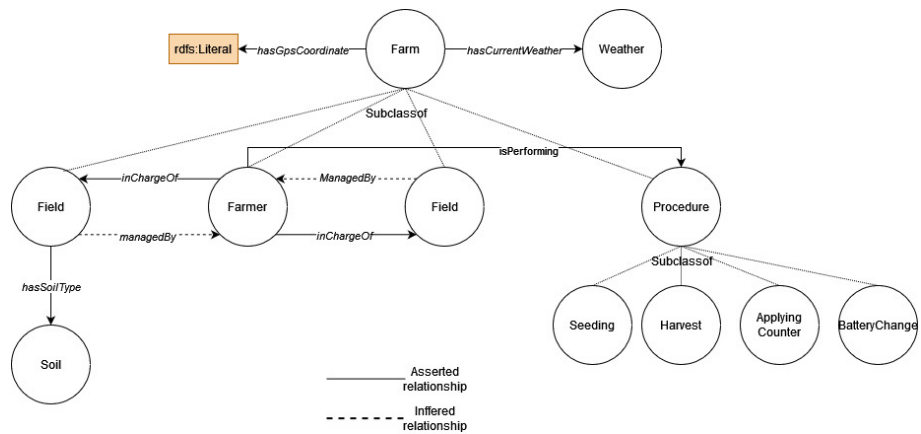


Figure 4.8: Ontological representation of the Farm class

4.3.4 . Rule Based System

As previously discussed, we have incorporated rules into our ontology to enhance its level of expressiveness. These rules have been specifically tailored for diverse applications, and their details are presented in the following Table 4.1. Each entry in the table comprises a description of the contextual situation in which the respective rule is intended to be applied, the corresponding literal expression formulated in Semantic Web Rule Language (SWRL), and an accompanying explanation of the rule's rationale and functionality.

Situation	SWRL	Description
Crops association	$\begin{aligned} & \text{canProvide}(\text{?p1}, \text{?x3}) \wedge \\ & \text{canRequire}(\text{?p2}, \text{?x3}) \Rightarrow \\ & \text{isGoodAssociation}(\text{?p1}, \text{?p2}) \end{aligned}$	<p>If one plant, denoted as p1, provides something that another plant, referred to as p2, requires, then this association between them can be deemed beneficial or favorable and they are linked through a goodAssociation property.</p>
Sensor Battery management	$\begin{aligned} & \text{hasLocation}(\text{?x1}, \text{?x2}) \\ & \wedge \text{inChargeOf}(\text{?x3}, \text{?x2}) \\ & \wedge \text{hasBatterylevel}(\text{?x1}, \text{?x4}) \\ & \wedge \text{?x4} \leq '20' \Rightarrow \\ & \text{isPerformingBattChange}(\text{?x3}, \text{?x1}) \end{aligned}$	<p>If a sensor, labeled as x1, is situated within a field, designated as x2, and a farmer, identified as x3, is responsible for the management of field x2, and the battery level of the sensor falls below or equals 20%, then a relationship denoted as 'isPerformingBatteryChange' is established between the farmer and the sensor.</p>
FUOTA procedure	$\begin{aligned} & \text{hasFirmware}(\text{?x1}, \text{?x2}) \\ & \wedge \text{isUpToDate}(\text{?x2}, \text{False}) \\ & \wedge \text{isUpToDate}(\text{?x3}, \text{True}) \\ & \Rightarrow \text{performFUOTA}(\text{?x1}, \text{?x3}) \end{aligned}$	<p>If a smart sensor, designated as x1, possesses a firmware labeled as x2 that is not up-to-date, and there exists a firmware x3 that is up-to-date, then a linkage is established between the sensor and firmware x3 for the purpose of Firmware Update Over-The-Air (FUOTA) procedure.</p>

Irrigation management	$ \begin{aligned} & hasWaterRequirement(?x1, y1) \wedge \\ & hasLocation(?x1, ?x2) \wedge \\ & hasLocation(?x3, ?x2) \wedge \\ & hasValue(?x3, y2) \leq y1 \wedge isActive(?x4, False) \wedge \\ & hasCurrentWeather(?x2, ?x5) \wedge ?x5 \neq \\ & \quad Storm' \vee Rain' \Rightarrow \\ & \quad isActive(?x4, True) \end{aligned} $	<p>If a plant, denoted as x1 and located in a field x2, has a water requirement represented by y1, and a sensor x3 situated in field x2 measures a water level denoted as y2 such that y2 is less than or equal to y1, and the weather conditions in field x2 are not characterized by rain or storms, and the irrigation valve x4 is already in an active state, then the directive is to activate the irrigation valve x4.</p>
Fan control	$ \begin{aligned} & hasLocation(?x1, ?x2) \wedge \\ & hasValue(?x1, y) \wedge \\ & hasLocation(?x3, ?x2) \wedge \\ & isActive(?x3, False) \wedge y \geq 30' \Rightarrow \\ & \quad isActive(?x3, True) \end{aligned} $	<p>If a sensor x1 within a greenhouse x2 detects a temperature exceeding 30 degrees, and if the fan x3 positioned in greenhouse x2 is not currently operational, it shall be activated.</p>
Pest and Disease detection and management	$ \begin{aligned} & hasLocation(?x1, ?x2) \wedge \\ & isDetecting(?x1, ?x3) \wedge \\ & hasValue(?x1, True) \wedge \\ & hasLocation(?x4, ?x2) \wedge \\ & hasCountermeasure(?x3, ?x5) \Rightarrow \\ & \quad isPerformingCtrMeasure(?x4, ?x3) \end{aligned} $	<p>If a Smart Sensor x1 located in a field x2 detects a threat x3 for which there exists a corresponding countermeasure x5, then the farmer x4 situated in field x2 is tasked with implementing the countermeasure x5.</p>

Table 4.1: SWRL rules

4.4 . Experimentation

In this section, we will populate our knowledge base with diverse instances representing different classes. Subsequently, we will conduct queries to demonstrate the system's capacity for interpreting a range of scenarios.

4.4.1 . Crops layout

The first scenario is to validate the identification of good associations between plants. We populated our ontology with the following plants along with their advantages and drawbacks according to Agroecology (Provide and Require):

- Tomatoes: Require protection against small insects, provide soil structure
- Corn: Provide vine support, provide shade to the ground
- Squash: Require nitrogen, require shade to the ground
- Peas: Require vine support, provide nitrogen
- Basil: Require soil structure, provide protection against small insects

We queried our ontology to select the best associations for tomatoes and Corn, and we obtained the following results:

- Tomatoes: Basil
- Corn: Squash, Peas

This showcases the effective architecture around the Provide and Require classes. This example is simple but illustrates the potential of such a method to model Agroecological knowledge.

4.4.2 . Sensors management

In our ontology, we have populated it with five sensors: s1, s2, s3, s4, and s5. Sensors s1, s2, and s3 are situated in Field1, managed by Farmer1, while sensors s4 and s5 are located in Field2, managed by Farmer2. Specifically, sensors s1 and s4 exhibit a battery level of 10%, while the remaining sensors maintain a battery level of 50%. We have the capability to query the system to retrieve information about sensors and farmers connected through the 'isImplementingBattChange' relationship. The query output yields s1 with Farmer1 and s4 with Farmer2, thereby demonstrating the successful establishment of associations between sensors and farmers within the knowledge base.

4.4.3 . Irrigation routine

For the irrigation management aspect, we have populated our knowledge base with various instances. Initially, we have introduced three types of crops: Potatoes with a water requirement of 20, tomatoes with a water requirement of 30, and watermelons with a requirement of 40. Subsequently, we have created two farm entities, namely, farm1 and farm2, each containing two fields (designated as f1, f2, f3, and f4). Within each field, we have positioned a humidity sensor (h1, h2, h3, h4) and an irrigation valve (v1, v2, v3, v4). We have configured the current weather conditions such that farm1 experiences sunny weather, while farm2 faces stormy weather. Furthermore, we have allocated the fields as follows: potatoes in fields f1 and f3, watermelons in f2, and tomatoes in f4. To complete the setup, we have set the soil moisture sensor values to 25. When querying for irrigation valves with an 'isActive' status set to True, the system outputs only valve v2, which corresponds to the watermelon field, in line with expectations. However, when the sensor values are adjusted to 15, the system outputs valves v1 and v2. Lastly, if we configure all the current weather conditions as 'Sunny,' the system identifies the need for irrigation for all crops, resulting in the activation of all irrigation valves

4.4.4 . Pest and disease detection and management

In this scenario, we enrich our ontology with the inclusion of smart sensors deployed in Field1 and Field2, each tasked with detecting slug pest invasions. Field1 is under the stewardship of Farmer1, while Field2 is managed by Farmer2. We introduce instances of tomato and salad plants. A relationship denoted as 'isSensitiveTo' is established between the salad plants and slug pests. Tomatoes are cultivated in Field1, while salad plants are grown in Field2. Upon setting the values of the smart sensors in both fields to 'true,' indicating the detection of slug pests, querying the system for fields with slug infestations yields Field1 and Field2 as results. Subsequently, querying for plants sensitive to slug pests in either Field1 or Field2 yields salad plants as the result. Finally, we inquire about the associated countermeasure for slug invasion, which is the application of lavender.

4.4.5 . Optimal seeding and harvest detection

We can also leverage the knowledge base to extract literal values and perform operations solely on them. For instance, we can inquire about the minimum temperature requirement for seeding a tomato plant and the geographical location of the farm. Utilizing these two pieces of information, we can develop a simple algorithm to compute the average daily temperature based on weather data at the farm's location. Subsequently, we can compare these temperature values with the minimum seeding temperature requirement (set at 15 degrees) to determine the optimal date for tomato planting. For this exercise, we accessed weather data from the Meteo France website for two

regions: Haut-de-France (Northern Region) and Occitanie (Southern Region). The computed average temperatures for both regions suggest that tomatoes can be planted as early as the beginning of April in Occitanie, whereas in Haut-de-France, one would need to wait until the end of May to meet the temperature requirements for tomato planting.

4.5 . Conclusion

This work on ontology development has been peer-reviewed and will be presented at the International conference KEOD (Knowledge Engineering and Ontology Development) that is taking place in Rome, Italy between the 15th and 18th of November 2023.

Our knowledge base proposes interesting results regarding our application. First of all, we have showcased that knowledge-based systems could offer compelling outputs regarding sensor management. By utilizing knowledge, a system can determine which smart sensors requires update. Additionally, the system is capable of detecting low battery levels in sensors and requesting that a farmer replace the battery by identifying which farmer is responsible for the sensors based on their location in the field. The system can also interpret data and propose effective actions regarding farm procedures such as seeding or implementing countermeasures to address threats. Finally, we have proposed an architecture for plant interaction based on providing and requiring classes. Each type of crop can provide certain advantages to the environment and require specific resources from it. By modeling these characteristics for each type of crop, the system can find suitable associations between crops following Agroecology principles.

Despite these promising results, we acknowledge that for our basic architecture to be truly effective, it should be enriched with more concepts provided by expert agroecological farmers, which would require more applied research. Moreover, we have demonstrated that existing knowledge can be reused for our application; therefore, efforts should be made to integrate more knowledge into it. Finally, the critical limitation identified is the need for the correct tools for experts to provide their knowledge to the system for later reuse, as well as the development of suitable interfacing tools for users to access this knowledge. Current manual methods, as we have used, are inefficient for retrieving vast amounts of knowledge, such as that in agriculture. Hopefully, new technologies such as Large Language Models offer new capabilities for human-machine interaction. These tools mimic human speech patterns; however, their responses are not solely based on knowledge but also on speech patterns. By combining knowledge-base systems with LLM query engines, we could develop systems that efficiently manage both explicit and implicit knowledge.

5 - General Conclusion

5.1 . General conclusion

Smart farming will play a significant part in the sustainable transformation of agriculture. With the advent of precision agriculture, thanks to IoT platforms, ensuring reliable communication among the myriad sensors and devices on farms becomes crucial. The development of intelligent sensors that embed AI algorithms for advanced calculations at the edge further amplifies this issue. Additionally, once data are collected, farmers often find it challenging to utilize them, especially in line with agroecology principles. In this context, this thesis introduces an energy-efficient protocol tailored for embedded AI sensors over the LoRa network and a knowledge-based system to aid farmers in decision-making based on sensor data and agroecology principles.

In Chapter One, we delve into the motivation and challenges of this work, addressing the issues of data communication and interpretation. Furthermore, this chapter explains the research methodology employed. Chapter 2 reviewed the current state-of-the-art IoT platforms for agricultural purposes, Embedded AI, and knowledge engineering for Agriculture. In Chapter 3, we proposed a model to meet Embedded AI sensors requirement in LoRa network by building our own smart devices and simulating their behavior regarding energy consumption. Chapter 4 elaborates into the notion of knowledge-based systems in the context of Agroecology and Smart Sensor management. It provides a comprehensive exploration of the ontology crafted for this study, its integral components, the logical rules that form the foundation of the intelligence layer, and the sequential steps involved in the inference process. Additionally, Chapter 4 presents a range of experiments designed to validate the efficacy of our systems.

Throughout the research process, multiple novel contributions have been made. Firstly, we proved the energy efficiency of Embedded AI systems for exploiting heavy data such as images or sound. Then we proposed a protocol to allow intelligent sensors to be remotely updated on LoRa low power wide area network, despite the heavy size of such firmware over the air update. Moreover, we proposed a novel energy-efficient approach to offload heavy sensor data, such as pictures or sounds, when the confidence score of a sensor drops below an acceptable threshold. This allows AI algorithms to be re-trained in the cloud with local data before being remotely updated. With the gathered data, we proposed a knowledge base created through ontology to model the different IoT devices and their interactions with plants. We also model agroecology principles in our knowledge base, making it an ontology

for the agroecological management of IoT platforms.

Overall our results showcase the future applications of Embedded AI in the agricultural domain and offer solutions to answer its networking issues with LoRa LPWAN network. Concurrently, it proved the efficiency of knowledge-base systems for complex application domains such as agroecology and sensors management.

5.2 . Future works

Despite the encouraging outcomes, we have identified certain limitations that should be the object of further research.

Firstly, on the networking side, even though we've addressed the challenges of transmitting large data files (such as firmware updates or images), LoRa technology—and broadly, all LPWAN protocols—are unsuitable for this type of application. We demonstrated that such transfers can occur sporadically in networks that aren't too dense, with a limited number of devices transmitting simultaneously. However, as the number of intelligent devices is anticipated to grow significantly, these wireless sensor networks are expected to become denser. This will challenge the ability of smart devices to communicate over LPWAN. To address these issues, several solutions can be explored. A straightforward remedy is to develop multi-Radio Access Technologies (multi-RAT), where each device is equipped with multiple network interfaces for different purposes. For instance, a sensor could have both a LoRa radio and a WiFi radio. The device would transmit its measurements and AI-inferred results via LoRa to conserve energy, while using WiFi for larger data transfers. This approach, however, complicates both the devices and the network infrastructure, as more specialized gateways are required. Another comprehensive solution involves harnessing the potential of future 6G networks. While the complete specifications and goals of 6G (the sixth generation of mobile networks) have not yet been fully defined or standardized, one primary objective will be to cater to the requirements of AI integration and denser IoT WSN networks. Consequently, we recommend that 6G provide devices the flexibility to interchangeably use either a specific low-power data channel for transmitting minimal data at a reduced energy cost or a higher data-rate channel for larger energy-costly data transfers. REWRITE:It is important to note also that future network systems based on satellite communication are becoming more prominent and might take a mor.

Secondly, regarding the knowledge representation and reasoning component, a major limitation exists in the acquisition of knowledge. Currently, our system represents the knowledge necessary for managing an IoT platform for Agroecology using a simple use-case ontology. We constructed different parts of this ontology incrementally, drawing solely on theoretical knowledge

and referencing existing ontologies, without striving for comprehensiveness. This method isn't well-suited to represent expansive domains such as agriculture, plant biology, and the broader food industry. One main challenge is the necessity of involving a vast panel of experts to integrate their knowledge into our ontology. Addressing this limitation requires research into improving the process of transferring knowledge from experts to machines. Emerging technologies like Large Language Models (LLMs) could provide valuable assistance. An LLM is an artificial intelligence model trained to understand and generate human-like text by processing vast amounts of diverse textual data. Such extensive training enables LLMs to identify a wide variety of linguistic patterns, equipping them for tasks like text generation, question answering, translation, and summarization. While models like OpenAI's GPT series showcase the potential of Large Language Models (LLMs), it's essential to acknowledge that their outputs are pattern-driven, rooted in training data, and devoid of genuine understanding or consciousness. Nevertheless, we assume that LLMs could play a pivotal role in the development of tools designed to assist experts in transferring their knowledge to a knowledge base through messaging apps and chatbot-based assistants.

5.3 . Peer-reviewed publication

The Doctoral school actively promotes its students to produce and share their contributions through scientific publications. In order to attain a PhD, students are required to generate a minimum of one peer-reviewed publication, although it is highly recommended to go beyond this requirement.

To date, I've had the honor of presenting two publications at international conferences. Presently, I have two additional papers that have been accepted and are scheduled for presentation in conferences – one in October and the other in November. Once the thesis writing process concludes, a Journal paper will encapsulate the comprehensive results achieved. Therefore As of now, there are a total of four produced peer-reviewed publications, with one more anticipated. Those are the following:

1. **TinyML Smart Sensor for Energy Saving in Internet of Things Precision Agriculture platform.** This paper was presented on the 13th International Conference on Ubiquitous and Future Networks, ICUFN. The conference took place in Barcelona, Spain, in July 2022. You can see a picture of me presenting in Figure 5.1.

DOI: 10.1109/ICUFN55119.2022.9829675



Figure 5.1: ICFUN presentation

2. **Energy efficient Firmware Over The Air Update for TinyML models in LoRaWAN agricultural networks.** This paper was presented on the 32nd International Telecommunication Networks and Applications Conference, ITNAC. The conference took place in Wellington, New-Zealand, in December 2022. You can see a picture of me presenting in Figure

DOI: 10.1109/ITNAC55475.2022.9998338



Figure 5.2: ITNAC presentation

3. **Embedded AI and Computation Offloading for 6G Green Communication.** This paper was presented at the International Conference on 6G Networking, 6GNet. The conference took place in Paris, France, in October 2023.



Figure 5.3: 6gnet presentation

4. **IoT-Enabled Agroecology: Advancing Sustainable Smart Farming through Knowledge-based Reasoning.** This paper was accepted for the 15th International Conference on Knowledge Engineering and Ontology Development, KEOD 2023. The conference is going to take place in Roma, Italy in November 2023.

5.4 . Additional training

The doctoral school of Paris-Saclay asks its PhD student to complete a certain number of mandatory additional training. This training are presented in table 5.1.

Activity type	Point to obtain	Note	Status
Ethic and scientific integrity course	2	MOOC proposed by the doctoral school	Done
Open science course	1	Class proposed by the doctoral school	Done
Sustainable development course	1	Class proposed by the doctoral school	Done
Scientific courses	10	1 point = 5h of science class	Done
Welcome day	1	In September	Done
Departure day	1	In May	Done

Table 5.1: Mandatory additional training of the Doctoral School of Paris-Saclay

- Ethic and scientific integrity course: The MOOC Ethic and stic is proposed by the doctoral school. You can find it at this address: <https://www.fun-mooc.fr/fr/cours/ethics-stics/>. It lasts approximately **15 hours**.
- Open science course: I attended the class of Open science: challenges and practice of opening up its publications and data on June 16, 2021. The Duration was **3 hours**;
- Sustainable development course: I attended the GEORGES BRAM Day 2023. 22nd day in History of Science and Epistemology and climate emergency on the January 13, 2023. It lasted **7 hours**..
- Scientific courses :
 - *Summer school in July 2022 located in Pisa : Enabling Technologies for Industrial Internet of Things*. The summer school is organized to provide 6 ECTS (about **54 hours** of front lessons (cf. the program in Appendice C)) in the fields of Electronics, RFID, Sensors, Networking, Web of Things, SW for IoT, Computer Engineering, targeting an audience of Bachelor (last year), Master and PhD students,

early stage researchers and practitioners from Academia or Industry. The summer school provides theoretical and practical lessons about recent advances in enabling technologies for electronics, electromagnetics, future wireless transceivers and wireless sensor networks, systems, edge and cloud computing, networks in scenarios such as IoT, Industry 4.0, Cyber-Physical Systems (CPS), autonomous vehicles, robots. The teaching material has been published in the IEEE book "Enabling Technologies for the Internet of Things: Wireless Circuits, Systems and Networks", 2018, Editor: Sergio Saponara, University of Pisa, Italy, ISBN: 9788793609747. You can find more information on the website of the school here : <https://www.unipi.it/index.php/engineering/item/6869-summer-school-internet-of-things>

- *MOOC Semantic Web technologies*: Online course about knowledge engineering, semantic web paradigm and ontology building. It lasted approximately **21 hours**. More information can be found on the page of the MOOC: <https://www.fun-mooc.fr/fr/cours/web-semantique-et-web-de-donnees/>
- *MOOC Introduction to Embedded Machine Learning*: This course give a broad overview of how machine learning works, how to train neural networks, and how to deploy those networks to microcontrollers, which is known as embedded machine learning or TinyML. This course represents approximately **17 hours** of work. More information can be found on the page of the MOOC : <https://www.coursera.org/learn/introduction-to-embedded-machine-learning>

The doctoral school also propose optional activity and training. Those are shown in Table 5.2.

Activity or training	Equivalence	Restrictions	Note	Point obtained
Language course	1 point = 5 hours	Maximum 5 point for all the PhD		0
Professionalizing class	1 point = 5 hours of classes		CV writing, learning specific IT tools, managing emotions, motivation, memory, mental representations, pedagogical alignment, etc.	3
Association participation	1 point per year of participation	Max 3 points for all the PhD.	You need to be part of the association board.	3
Commitment to boards, committees, juries (outside teaching)	1 point = 5h	Maximum 3 point for all the PhD		1
Teaching	1 point = 19h of classes	Maximum 5 point for all the Ph.		5
International mobility	3 points for 4 weeks of mobility.	Maximum 3 point for all the PhD.		0
Supervision or co-supervision of interns		Maximum 1 point		1
Scientific courses	1 point = 5h of classes		In addition of the mandatory ones.	4

Table 5.2: Optional activities and trainings

For now I completed the following optional training and activity :

- Professionalizing class:
 - *Writing and publishing research paper course: An introduction course*

to give advices and good practices to the doctoral students in order to write research papers. This was **4.5 hours** long.

- *Korsa programm*: This program aimed to help students manage stress and anxiety throughout their PhD. It was **15 hours** long.

In total those training accounted for **20 hours** and **4 points**.

- Association Participation: For **5 years** now (since 2018), I have been part of the association for robotics and electronics Eceborg. This association aim to help student in their project and manage bigger one such as the french cup of robotics. You can find more information on the website: <https://eceborg.fr/>. I am personally in charge of the vegetable garden on top of the school. I participate in the development of system to automate the garden such as weather station, agricultural robot, solar energy experiences, irrigation management, greenhouse climate control and others. I'm also responsible for the maker area meaning making sure the 3D printer fleet is up and running and performing maintenance on them. In Figure 5.4 you can see me and the member of the association when I was its president in 2019 and you can see the vegetable garden in its current state in Figure 5.5. The overall participation account for **3 points** as I was part of the board for more than 3 years.



Figure 5.4: Eceborg group photo when I was president

- Commitment to boards, committees, juries (outside teaching): I took part in some of the committees for student admission of my school. I also was asked to participate to juries for student project. Finally I also correct internship report of master student. In total, those activity account for **5 hours** meaning **1 point** gained.



Figure 5.5: Vegetable garden of the school

- Teaching: As stated in my contract, it was mandatory for me to teach 100 hours per year. During my three years at ECE, I did those **300 hours** in the form of practical work classes and group project supervision. I gave practical works about FPGA design (*Field Programmable Gate Array*), basic electronic systems and programming (C++ classes). I also supervised one project of student in my last year where they had to build a seeding greenhouse with climate control. In Figure 5.6, you can see me giving a class in front of students. According to the doctoral school, those teaching hours allow me an extra **5 points**.



Figure 5.6: Class teaching

- Supervision or co-supervision of interns: I helped co-supervised an intern who was working on the green-panel project of the school. This project aims to prove the lowering influence on temperature of different layers of plants growing under solar panel to increase their efficiency. I helped the intern deploy a fleet of sensors of the different panel to monitor their temperature throughout the experience. This

account for **1 point**.

- Scientific courses: As stated above in the mandatory training part, I attended a total of 92 hours of classes. 50 hours are mandatory only meaning I did an extra **42 hours**. This account for **4 points**.

In conclusion, the Doctoral School require a total of 180 points to obtain the PhD and I obtained 189. The thesis and research work account for 158 points. The total of additional training and activities (mandatory and optional) is 31 points (14 mandatory and 17 optional).

5.5 . Future opportunities

As I approach the end of my student journey, a decade after obtaining my high school diploma, I look back with nostalgia. I truly cherished those years dedicated to acquiring knowledge. However, I'm equally enthusiastic about the future, and I'm already exploring opportunities for the next phase of my career. I aspire to remain in the research field, either in academia or on a more industry-oriented trajectory. Throughout my studies, I had the opportunity to connect with many professionals in the smart farming domain, and I'm currently in discussions for a position in a research and development project at the Agrotechnopôle of INRAE (Institut National de Recherche Agronomique et Environnementale), the French research center for agriculture.

In conclusion, I want to thank again all the people who made this thesis possible. I also want to thank the reader for taking the time to explore my research.

6 - Annexe : Résumé de thèse en Français

L'agriculture moderne nécessite une profonde transformation pour répondre aux défis du développement durable tout en nourrissant qualitativement et quantitativement la population mondiale croissante. En effet, celle-ci repose sur l'utilisation d'éléments chimiques polluant les sols et présents en quantités limitées sur terre. L'usage de machinerie lourde nécessitant des combustibles fossiles et tassant le sol limite sa fertilité à long terme. Enfin, ces deux méthodes sont le pilier de la monoculture standardisée qui consiste à cultiver un seul type de plantes sur de vastes territoires aux écosystèmes et conditions climatiques différents, facilitant le développement de maladies et la déplétion en nutriments des sols. Pour répondre à ces problématiques, les agriculteurs adoptent le "Smart Farming" ou agriculture intelligente. C'est une méthode agricole qui utilise les nouvelles technologies de l'information et de la communication pour améliorer l'efficacité, la productivité et la durabilité de la production agricole. Elle englobe l'usage de capteurs, l'internet des objets (IoT : Internet of Things), l'intelligence artificielle (IA), l'analyse de données, la robotique et divers autres outils numériques optimisant des aspects tels que la gestion des sols, l'irrigation, la lutte antiparasitaire ou encore la gestion de l'élevage. L'objectif est d'augmenter la production tout en réduisant la consommation de ressources, minimisant les déchets et améliorant la qualité des produits. Toutefois, malgré ses avantages et son déploiement réussi dans divers projets, l'agriculture intelligente rencontre des limites : Tout d'abord, les solutions de SFs doivent être adaptées économiquement au domaine agricole où les marges sont faibles. Ensuite, les fermiers doivent être accompagnés pour l'implémentation et l'utilisation de ces nouvelles méthodes. Enfin, le SF est trop souvent employé pour rendre des méthodes d'agriculture polluante plus durables au lieu de proposer des solutions pour rendre l'agriculture durable plus productive. Une des méthodes durables les plus connues est l'agroécologie qui se base sur l'observation et l'interprétation de l'écosystème local pour adapter les pratiques agricoles à celui-ci. Dans ce contexte, cette thèse s'est intéressée à la création d'une plateforme IoT pour les fermiers afin de faciliter leur transition écologique. Une plateforme IoT est un système informatique capable de percevoir des données dans l'environnement (notamment à l'aide de capteurs), de les interpréter et de proposer des actions à réaliser en fonction des données récoltées. Un exemple serait un capteur d'humidité du sol qui détermine la quantité d'eau disponible pour les plantes proches et déciderait d'ouvrir ou non une vanne d'irrigation pour fournir la quantité précise en eau tout en évitant de gaspiller la ressource. Dans la réalité, l'irrigation dépend de bien plus de phénomènes comme le type de sol et de plantes, la température, les prévisions météo, la date, etc. Cette thèse s'est

intéressée à deux problématiques de telles plateformes IoT : la perception de phénomènes complexes grâce à l'IA embarquée ainsi que la prise de décision à l'aide de méthodes d'ingénierie des connaissances. L'IA embarquée, également appelée "Embedded AI" en anglais, désigne l'intégration de technologies et d'algorithmes d'intelligence artificielle dans des dispositifs ou des systèmes informatiques avec des puissances de calcul restreintes tels que des capteurs. L'avantage de ces capteurs est leur indépendance vis-à-vis du cloud pour interpréter des données complexes comme les images. Cette indépendance permet d'obtenir des gains en termes de données, de confidentialité et surtout en termes d'énergie, car le transfert de données est généralement le processus le plus coûteux énergétiquement, ce qui pose problème pour des capteurs opérant sur batterie. Grâce à l'IA embarquée, une flotte de capteurs peut détecter des phénomènes complexes en agroécologie comme le développement de maladies sur les feuilles, le nombre de fruits prêts à la récolte, etc. Le problème de cette méthode est que le modèle d'IA embarqué doit être mis à jour fréquemment, ce qui nécessite de collecter des données dans l'environnement où le capteur est déployé afin de ré-entraîner des modèles dans le Cloud pour les redéployer à distance par la suite. Or les capteurs agricoles utilisent des réseaux de communication de type LPWAN (Low Power Wide Area Network - Réseau étendu à faible consommation d'énergie) qui permettent aux capteurs de communiquer sur de longues distances avec une très faible consommation d'énergie, mais au détriment du débit de données. Cette thèse a donc proposé des protocoles algorithmiques de mise à jour à distance et de collecte ponctuelle de données pour des modèles d'intelligence artificielle fonctionnant sur des réseaux LoRa. L'efficacité énergétique de ces systèmes a été prouvée ainsi que leur faisabilité, cependant il a été déterminé que la mise à l'échelle de telles solutions est complexe, ainsi les futurs réseaux de type 6G devront adresser cette problématique. Cette partie a fait l'objet de 3 publications dans des conférences internationales. La deuxième contribution de cette thèse concerne la prise de décisions à partir de ces données perçues. Pour ce faire, nous avons modélisé les connaissances agroécologiques ainsi que les caractéristiques des différents capteurs afin de proposer un système capable de proposer des plans pour des fermes en multicultures ainsi que la gestion de l'irrigation et des menaces (maladies et parasites). Nous avons utilisé des méthodes sémantiques basées sur les ontologies pour réaliser cela. Cette partie a fait l'objet d'une publication dans une conférence internationale. Dans l'ensemble, nos résultats mettent en avant les futures applications de l'IA Embarquée dans le domaine agricole et proposent des solutions pour répondre à ses problèmes de mise en réseau avec le réseau LoRa LPWAN. Parallèlement, a été démontré l'efficacité des systèmes de base de connaissances pour des domaines d'application complexes tels que l'agroécologie et la gestion des capteurs.

Bibliography

- [1] Giovanni Federico. *Feeding the world: an economic history of agriculture, 1800-2000*. Princeton University Press, 2010.
- [2] Jonathan A Foley et al. "Solutions for a cultivated planet." In: *Nature* 478.7369 (2011), pp. 337–342.
- [3] H Charles J Godfray et al. "Food security: the challenge of feeding 9 billion people." In: *science* 327.5967 (2010), pp. 812–818.
- [4] David Tilman et al. "Global food demand and the sustainable intensification of agriculture." In: *Proceedings of the national academy of sciences* 108.50 (2011), pp. 20260–20264.
- [5] Lu Zhang et al. "The impact of agricultural chemical inputs on environment: global evidence from informetrics analysis and visualization." In: *International Journal of low-Carbon technologies* 13.4 (2018), pp. 338–352.
- [6] Polyxeni Nicolopoulou-Stamati et al. "Chemical pesticides and human health: the urgent need for a new concept in agriculture." In: *Frontiers in public health* 4 (2016), p. 148.
- [7] MA Hamza and Walter K Anderson. "Soil compaction in cropping systems: A review of the nature, causes and possible solutions." In: *Soil and tillage research* 82.2 (2005), pp. 121–145.
- [8] Hailu Gebru. "A review on the comparative advantages of intercropping to mono-cropping system." In: *Journal of Biology, Agriculture and Healthcare* 5.9 (2015), pp. 1–13.
- [9] Deborah Lawrence and Karen Vandecar. "Effects of tropical deforestation on climate and agriculture." In: *Nature climate change* 5.1 (2015), pp. 27–36.
- [10] R Lal. "Tillage and agricultural sustainability." In: *Soil and tillage research* 20.2-4 (1991), pp. 133–146.
- [11] Hugh Turrall, Jacob Burke, Jean-Marc Faurès, et al. *Climate change, water and food security*. 36. Food and agriculture organization of the United nations (FAO), 2011.
- [12] Veronique Beckers et al. "The impact of urbanization on agricultural dynamics: A case study in Belgium." In: *Journal of Land Use Science* 15.5 (2020), pp. 626–643.

- [13] D Mishra et al. "Role of bio-fertilizer in organic agriculture: a review." In: *Research Journal of Recent Sciences ISSN 2277* (2013), p. 2502.
- [14] RG McLaren and CJ Smith. "Issues in the disposal of industrial and urban wastes." In: *Contaminants and the Soil Environment in the Australasia-Pacific Region: Proceedings of the First Australasia-Pacific Conference on Contaminants and Soil Environment in the Australasia-Pacific Region, held in Adelaide, Australia, 18–23 February 1996*. Springer. 1996, pp. 183–212.
- [15] WN Smith et al. "Crop residue removal effects on soil carbon: Measured and inter-model comparisons." In: *Agriculture, ecosystems & environment* 161 (2012), pp. 27–38.
- [16] Dean R Paini et al. "Global threat to agriculture from invasive species." In: *Proceedings of the National Academy of Sciences* 113.27 (2016), pp. 7575–7579.
- [17] John T Hardy. *Climate change: causes, effects, and solutions*. John Wiley & Sons, 2003.
- [18] Tim Wheeler and Joachim Von Braun. "Climate change impacts on global food security." In: *Science* 341.6145 (2013), pp. 508–513.
- [19] Christoph Müller et al. "Climate change impacts on agricultural yields." In: (2010).
- [20] KC Samir and Wolfgang Lutz. "The human core of the shared socioeconomic pathways: Population scenarios by age, sex and level of education for all countries to 2100." In: *Global Environmental Change* 42 (2017), pp. 181–192.
- [21] Janet Ranganathan et al. "How to sustainably feed 10 billion people by 2050, in 21 charts." In: (2018).
- [22] Benjamin Plackett. "Changing diets at scale." In: *Nature* 588.7837 (2020), S70–S70.
- [23] Paul Harrison. *World agriculture: towards 2015/2030. Summary report*. FAO, 2002.
- [24] D John Shaw. "World food summit, 1996." In: *World Food Security: A History since 1945*. Springer, 1996, pp. 347–360.
- [25] Alexandre Meybeck and Vincent Gitz. "Sustainable diets within sustainable food systems." In: *Proceedings of the Nutrition Society* 76.1 (2017), pp. 1–11.
- [26] Hannah Ritchie and Max Roser. "Land Use." In: *Our World in Data* (2013). <https://ourworldindata.org/land-use>.

- [27] Guancheng Guo, Qiyu Wen, and Jingjuan Zhu. "The impact of aging agricultural labor population on farmland output: from the perspective of farmer preferences." In: *Mathematical problems in Engineering* 2015 (2015), pp. 1–7.
- [28] Carlo Giua, Valentina Cristiana Materia, and Luca Camanzi. "Smart farming technologies adoption: Which factors play a role in the digital transition?" In: *Technology in Society* 68 (2022), p. 101869.
- [29] Klaus Deininger. "Challenges posed by the new wave of farmland investment." In: *The journal of peasant studies* 38.2 (2011), pp. 217–247.
- [30] Rovshen Ishangulyyev, Sanghyo Kim, and Sang Hyeon Lee. "Understanding food loss and waste—why are we losing and wasting food?" In: *Foods* 8.8 (2019), p. 297.
- [31] Wayne Wakeland, Susan Cholette, and Kumar Venkat. "Food transportation issues and reducing carbon footprint." In: *Green technologies in food production and processing* (2012), pp. 211–236.
- [32] World Health Organization et al. *The state of food security and nutrition in the world 2019: safeguarding against economic slowdowns and downturns*. Vol. 2019. Food & Agriculture Org., 2019.
- [33] Achim Walter et al. "Smart farming is key to developing sustainable agriculture." In: *Proceedings of the National Academy of Sciences* 114.24 (2017), pp. 6148–6150.
- [34] Ye Liu et al. "From Industry 4.0 to Agriculture 4.0: Current status, enabling technologies, and research challenges." In: *IEEE Transactions on Industrial Informatics* 17.6 (2020), pp. 4322–4334.
- [35] Jayavardhana Gubbi et al. "Internet of Things (IoT): A vision, architectural elements, and future directions." In: *Future Generation Computer Systems* 29.7 (2013), pp. 1645–1660. issn: 0167-739X. doi: <https://doi.org/10.1016/j.future.2013.01.010>. url: <https://www.sciencedirect.com/science/article/pii/S0167739X13000241>.
- [36] Godwin Idoje, Tasos Dagiuklas, and Muddesar Iqbal. "Survey for smart farming technologies: Challenges and issues." In: *Computers & Electrical Engineering* 92 (2021), p. 107104.
- [37] Ramon Sanchez-Iborra and Antonio F Skarmeta. "Tinyml-enabled frugal smart objects: Challenges and opportunities." In: *IEEE Circuits and Systems Magazine* 20.3 (2020), pp. 4–18.

- [38] Massimo Merenda, Carlo Porcaro, and Demetrio Iero. "Edge machine learning for ai-enabled iot devices: A review." In: *Sensors* 20.9 (2020), p. 2533.
- [39] Shubhangi K Gawali and Mukund K Deshmukh. "Energy autonomy in IoT technologies." In: *Energy Procedia* 156 (2019), pp. 222–226.
- [40] Partha Pratim Ray. "A review on TinyML: State-of-the-art and prospects." In: *Journal of King Saud University-Computer and Information Sciences* (2021).
- [41] Kais Mekki et al. "A comparative study of LPWAN technologies for large-scale IoT deployment." In: *ICT express* 5.1 (2019), pp. 1–7.
- [42] Haider Mahmood Jawad et al. "Energy-efficient wireless sensor networks for precision agriculture: A review." In: *Sensors* 17.8 (2017), p. 1781.
- [43] Wim JC Verhagen et al. "A critical review of Knowledge-Based Engineering: An identification of research challenges." In: *Advanced Engineering Informatics* 26.1 (2012), pp. 5–15.
- [44] Charith Perera et al. "Context aware computing for the internet of things: A survey." In: *IEEE communications surveys & tutorials* 16.1 (2013), pp. 414–454.
- [45] Lorena Otero-Cerdeira, Francisco J Rodriguez-Martinez, and Alma Gómez-Rodríguez. "Ontology matching: A literature review." In: *Expert Systems with Applications* 42.2 (2015), pp. 949–971.
- [46] JM McKinion and HE Lemmon. "Expert systems for agriculture." In: *Computers and Electronics in Agriculture* 1.1 (1985), pp. 31–40.
- [47] Seline S Meijer et al. "The role of knowledge, attitudes and perceptions in the uptake of agricultural and agroforestry innovations among smallholder farmers in sub-Saharan Africa." In: *International journal of agricultural sustainability* 13.1 (2015), pp. 40–54.
- [48] Kirtan Jha et al. "A comprehensive review on automation in agriculture using artificial intelligence." In: *Artificial Intelligence in Agriculture* 2 (2019), pp. 1–12.
- [49] Laurens Klerkx, Emma Jakku, and Pierre Labarthe. "A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda." In: *NJAS-Wageningen journal of life sciences* 90 (2019), p. 100315.

- [50] Emma Jakku et al. "“If they don’t tell us what they do with it, why would we trust them?” Trust, transparency and benefit-sharing in Smart Farming." In: *NJAS-Wageningen Journal of Life Sciences* 90 (2019), p. 100285.
- [51] Dieisson Pivoto et al. "Scientific development of smart farming technologies and their application in Brazil." In: *Information processing in agriculture* 5.1 (2018), pp. 21–32.
- [52] Kyle Wiens. "New High-Tech Farm Equipment Is a Nightmare for Farmers." In: *Wired Business* (2015). url: <https://www.wired.com/2015/02/new-high-tech-farm-equipment-nightmare-farmers/>.
- [53] Callum Eastwood et al. "Managing socio-ethical challenges in the development of smart farming: from a fragmented to a comprehensive approach for responsible research and innovation." In: *Journal of agricultural and environmental ethics* 32.5-6 (2019), pp. 741–768.
- [54] P Waldman and L Mulvany. *Farmers Fight John Deere Over Who Gets to Fix an \$800,000 Tractor*. Bloomberg. 2020.
- [55] Spyros Fountas et al. "Agricultural robotics for field operations." In: *Sensors* 20.9 (2020), p. 2672.
- [56] Irina Toma et al. "Small-scale farming and food security–policy perspectives from central and eastern europe." In: *Global Food Security* 29 (2021), p. 100504.
- [57] "Ending hunger: science must stop neglecting smallholder farmers." In: *Nature* 586.336 (2020), p. 336. doi: [doi:https://doi.org/10.1038/d41586-020-02849-6](https://doi.org/10.1038/d41586-020-02849-6).
- [58] Zeynab Jouzi et al. "Organic farming and small-scale farmers: Main opportunities and challenges." In: *Ecological economics* 132 (2017), pp. 144–154.
- [59] Sarah K Lowder, Jakob Skoet, and Terri Raney. "The number, size, and distribution of farms, smallholder farms, and family farms worldwide." In: *World Development* 87 (2016), pp. 16–29.
- [60] Steve Wiggins, Johann Kirsten, and Luis Llambi. "The future of small farms." In: *World development* 38.10 (2010), pp. 1341–1348.
- [61] Pierre-Marie Bosc et al. *Investing in smallholder agriculture for food security*. Tech. rep. HLPE, 2013.
- [62] Gerard D’souza and John Ikerd. "Small farms and sustainable development: is small more sustainable?" In: *Journal of agricultural and applied economics* 28.1 (1996), pp. 73–83.

- [63] "Ruralization project." In: (2022). url: <https://ruralization.eu/>.
- [64] Siti Fatimahwati Pehin Dato Musa and Khairul Hidayatullah Basir. "Smart farming: towards a sustainable agri-food system." In: *British Food Journal* 123.9 (2021), pp. 3085–3099.
- [65] Alexander Wezel et al. "Agroecology as a science, a movement and a practice. A review." In: *Agronomy for sustainable development* 29 (2009), pp. 503–515.
- [66] Miguel A Altieri. *Agroecology: the science of sustainable agriculture*. CRC Press, 2018.
- [67] Patricia A Terry, Debra Pearson, and Gregory Holder. "Sustainable Agriculture: Nutrition of Indigenous American 3 Sisters Garden Compared to Monoculture Corn Production and a Cool Old Squash." In: *Journal of Scientific Research and Reports* 26.8 (2020), pp. 99–108.
- [68] Joyce E Parker et al. "Companion planting and insect pest control." In: *Weed and pest control-conventional and new challenges* 10 (2013), p. 55044.
- [69] Sjaak Wolfert, Daan Goense, and Claus Aage Grøn Sørensen. "A Future Internet Collaboration Platform for Safe and Healthy Food from Farm to Fork." In: *2014 Annual SRII Global Conference*. 2014, pp. 266–273. doi: [10.1109/SRII.2014.47](https://doi.org/10.1109/SRII.2014.47).
- [70] Kevin Ashton et al. "That 'internet of things' thing." In: *RFID journal* 22.7 (2009), pp. 97–114.
- [71] Rafiullah Khan et al. "Future internet: the internet of things architecture, possible applications and key challenges." In: *2012 10th international conference on frontiers of information technology*. IEEE. 2012, pp. 257–260.
- [72] Subramanian Balaji, Karan Nathani, and Rathnasamy Santhakumar. "IoT technology, applications and challenges: a contemporary survey." In: *Wireless personal communications* 108 (2019), pp. 363–388.
- [73] Mohamed K Yousef et al. *Stress physiology in livestock. Volume I. Basic principles*. CRC press, 1985.
- [74] Federico Civerchia et al. "Industrial Internet of Things monitoring solution for advanced predictive maintenance applications." In: *Journal of Industrial Information Integration* 7 (2017), pp. 4–12.

- [75] Roberto Minerva, Gyu Myoung Lee, and Noel Crespi. "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models." In: *Proceedings of the IEEE* 108.10 (2020), pp. 1785–1824.
- [76] Marco Lombardi, Francesco Pascale, and Domenico Santaniello. "Internet of things: A general overview between architectures, protocols and applications." In: *Information* 12.2 (2021), p. 87.
- [77] Ala Al-Fuqaha et al. "Internet of things: A survey on enabling technologies, protocols, and applications." In: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.
- [78] Partha Pratim Ray. "A survey on Internet of Things architectures." In: *Journal of King Saud University-Computer and Information Sciences* 30.3 (2018), pp. 291–319.
- [79] Alem Čolaković and Mesud Hadžialić. "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues." In: *Computer networks* 144 (2018), pp. 17–39.
- [80] Mohammed Riyadh Abdmeziem, Djamel Tandjaoui, and Imed Romdhani. "Architecting the internet of things: state of the art." In: *Robots and Sensor Clouds* (2016), pp. 55–75.
- [81] Transforma Insights. *Global iot market to grow to 24.1 billion devices in 2030, generating 1.5trillionannualrevenue*. 2020. url: <https://transformainsights.com/news/iot-market-24-billion-usd15-trillion-revenue-2030>.
- [82] Lin William Cong, Beibei Li, and Qingquan Tony Zhang. "Internet of Things: Business Economics and Applications." In: *Review of business* 41.1 (2021).
- [83] Amjad Rehman et al. "A revisit of internet of things technologies for monitoring and control strategies in smart agriculture." In: *Agronomy* 12.1 (2022), p. 127.
- [84] Olakunle Elijah et al. "An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges." In: *IEEE Internet of things Journal* 5.5 (2018), pp. 3758–3773.
- [85] Antonis Tzounis et al. "Internet of Things in agriculture, recent advances and future challenges." In: *Biosystems engineering* 164 (2017), pp. 31–48.
- [86] Muhammad Shoaib Farooq et al. "A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming." In: *IEEE Access* 7 (2019), pp. 156237–156271.

- [87] Steffen Bauer and Lindsay C Stringer. "The role of science in the global governance of desertification." In: *The Journal of Environment & Development* 18.3 (2009), pp. 248–267.
- [88] Saskia Van der Kooij et al. "The efficiency of drip irrigation unpacked." In: *Agricultural Water Management* 123 (2013), pp. 103–110.
- [89] Kanchana Rajaram and R Sundareswaran. "IoT based crop-field monitoring and precise irrigation system using crop water requirement." In: *International Conference on Computational Intelligence in Data Science*. Springer, 2020, pp. 291–304.
- [90] Karan Kansara et al. "Sensor based automated irrigation system with IOT: A technical review." In: *International Journal of Computer Science and Information Technologies* 6.6 (2015), pp. 5331–5333.
- [91] S Jain and KS Vani. "A survey of the automated irrigation systems and the proposal to make the irrigation system intelligent." In: *Int. J. Comput. Sci. Eng* 6 (2018), pp. 357–360.
- [92] Priyamitra Munoth, Rohit Goyal, and Kuldeep Tiwari. "Sensor based irrigation system: A review." In: *NCACE. USA* (2016).
- [93] Laura Garcia et al. "IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture." In: *Sensors* 20.4 (2020), p. 1042.
- [94] Raveendra Kumar Rai, Vijay P Singh, and Alka Upadhyay. *Planning and evaluation of irrigation projects: methods and implementation*. Academic press, 2017.
- [95] Fidaus Kamaruddin et al. "IoT-based intelligent irrigation management and monitoring system using arduino." In: *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 17.5 (2019), pp. 2378–2388.
- [96] R Santhana Krishnan et al. "Fuzzy logic based smart irrigation system using internet of things." In: *Journal of Cleaner Production* 252 (2020), p. 119902.
- [97] John L Havlin. "Soil: Fertility and nutrient management." In: *Landscape and land capacity*. CRC Press, 2020, pp. 251–265.
- [98] Rattan Lal. "Soil structure and sustainability." In: *Journal of sustainable agriculture* 1.4 (1991), pp. 67–92.
- [99] Emanuel Epstein and Arnold J Bloom. *Mineral nutrition of plants: principles and perspectives*. Sinauer, 1853.

- [100] Simona Bungau et al. "Expatriating the impact of anthropogenic aspects and climatic factors on long-term soil monitoring and management." In: *Environmental Science and Pollution Research* 28.24 (2021), pp. 30528–30550.
- [101] SN Shylaja and MB Veena. "Real-time monitoring of soil nutrient analysis using WSN." In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE. 2017, pp. 3059–3062.
- [102] G Sai Pravallika et al. "Proficient smart soil based IoT system for crop prediction." In: *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE. 2020, pp. 752–757.
- [103] R Madhumathi, T Arumuganathan, and R Shruthi. "Soil NPK and moisture analysis using wireless sensor networks." In: *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE. 2020, pp. 1–6.
- [104] Bhuvaneswari Swaminathan et al. "IoT-driven artificial intelligence technique for fertilizer recommendation model." In: *IEEE Consumer Electronics Magazine* 12.2 (2022), pp. 109–117.
- [105] T Blesslin Sheeba et al. "Machine Learning Algorithm for Soil Analysis and Classification of Micronutrients in IoT-Enabled Automated Farms." In: *Journal of Nanomaterials* 2022 (2022).
- [106] Narendra Khatri et al. "An IoT-based innovative real-time pH monitoring and control of municipal wastewater for agriculture and gardening." In: *Proceedings of First International Conference on Smart System, Innovations and Computing: SSIC 2017, Jaipur, India*. Springer. 2018, pp. 353–362.
- [107] George Archbold Taylor et al. "pH measurement IoT system for precision agriculture applications." In: *IEEE Latin America Transactions* 17.05 (2019), pp. 823–832.
- [108] A Hemmat and VI Adamchuk. "Sensor systems for measuring soil compaction: Review and analysis." In: *Computers and electronics in agriculture* 63.2 (2008), pp. 89–103.
- [109] Muthumanickam Dhanaraju et al. "Smart farming: Internet of Things (IoT)-based sustainable agriculture." In: *Agriculture* 12.10 (2022), p. 1745.
- [110] Marcello Donatelli et al. "Modelling the impacts of pests and diseases on agricultural systems." In: *Agricultural systems* 155 (2017), pp. 213–224.

- [111] Carolyn Dimitri, Anne Effland, and Neilson C Conklin. *The 20th century transformation of US agriculture and farm policy*. Tech. rep. 2005.
- [112] Jun Liu and Xuewei Wang. "Plant diseases and pests detection based on deep learning: a review." In: *Plant Methods* 17 (2021), pp. 1–18.
- [113] Tiago Domingues, Tomás Brandão, and João C Ferreira. "Machine learning for detection and prediction of crop diseases and pests: A comprehensive survey." In: *Agriculture* 12.9 (2022), p. 1350.
- [114] Jingcheng Zhang et al. "Monitoring plant diseases and pests through remote sensing technology: A review." In: *Computers and Electronics in Agriculture* 165 (2019), p. 104943.
- [115] Suyash S Patil and Sandeep A Thorat. "Early detection of grapes diseases using machine learning and IoT." In: *2016 second international conference on Cognitive Computing and Information Processing (CCIP)*. IEEE. 2016, pp. 1–5.
- [116] Sergio Trilles et al. "Development of an open sensorized platform in a smart agriculture context: A vineyard support system for monitoring mildew disease." In: *Sustainable Computing: Informatics and Systems* 28 (2020), p. 100309.
- [117] Sehan Kim, Meonghun Lee, and Changsun Shin. "IoT-based strawberry disease prediction system for smart farming." In: *Sensors* 18.11 (2018), p. 4051.
- [118] Apeksha Thorat, Sangeeta Kumari, and Nandakishor D Valakunde. "An IoT based smart solution for leaf disease detection." In: *2017 international conference on big data, IoT and data science (BIGD)*. IEEE. 2017, pp. 193–198.
- [119] Shufen Zhang, Xuebin Chen, and Shi Wang. "Research on the monitoring system of wheat diseases, pests and weeds based on IOT." In: *2014 9th International Conference on Computer Science & Education*. IEEE. 2014, pp. 981–985.
- [120] Heamin Lee et al. "Disease and pest prediction IoT system in orchard: A preliminary study." In: *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE. 2017, pp. 525–527.
- [121] Nidhi Kundu et al. "IoT and interpretable machine learning based framework for disease prediction in pearl millet." In: *Sensors* 21.16 (2021), p. 5386.

- [122] Ahmed Khattab et al. "An IoT-based cognitive monitoring system for early plant disease forecast." In: *Computers and Electronics in Agriculture* 166 (2019), p. 105028.
- [123] Demin Gao et al. "A framework for agricultural pest and disease monitoring based on internet-of-things and unmanned aerial vehicles." In: *Sensors* 20.5 (2020), p. 1487.
- [124] Philipp Lottes et al. "UAV-based crop and weed classification for smart farming." In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3024–3031.
- [125] Nuttakarn Kitpo and Masahiro Inoue. "Early rice disease detection and position mapping system using drone and IoT architecture." In: *2018 12th South East Asian Technical University Consortium (SEATUC)*. Vol. 1. IEEE. 2018, pp. 1–5.
- [126] Achilles D Boursianis et al. "Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review." In: *Internet of Things* 18 (2022), p. 100187.
- [127] Abhijit Khadatkhar, CR Mehta, and CP Sawant. "Application of robotics in changing the future of agriculture." In: (2022).
- [128] Xuewei Wang, Jun Liu, and Xiaoning Zhu. "Early real-time detection algorithm of tomato diseases and pests in the natural environment." In: *Plant Methods* 17.1 (2021), pp. 1–17.
- [129] Ilyas Potamitis, Panagiotis Eliopoulos, and Iraklis Rigakis. "Automated remote insect surveillance at a global scale and the internet of things." In: *Robotics* 6.3 (2017), p. 19.
- [130] Nipuna Chamara et al. "Ag-IoT for crop and environment monitoring: Past, present, and future." In: *Agricultural systems* 203 (2022), p. 103497.
- [131] DK Sreekantha and AM Kavya. "Agricultural crop monitoring using IOT-a study." In: *2017 11th International conference on intelligent systems and control (ISCO)*. IEEE. 2017, pp. 134–139.
- [132] Carlos Cambra et al. "An IoT service-oriented system for agriculture monitoring." In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–6.
- [133] Yunong Tian et al. "Apple detection during different growth stages in orchards using the improved YOLO-V3 model." In: *Computers and electronics in agriculture* 157 (2019), pp. 417–426.
- [134] Peiyuan Jiang et al. "A Review of Yolo algorithm developments." In: *Procedia Computer Science* 199 (2022), pp. 1066–1073.

- [135] Toby N Carlson and David A Ripley. "On the relation between NDVI, fractional vegetation cover, and leaf area index." In: *Remote sensing of Environment* 62.3 (1997), pp. 241–252.
- [136] Marek Wójtowicz, Andrzej Wójtowicz, Jan Piekarczyk, et al. "Application of remote sensing methods in agriculture." In: *Communications in biometry and crop science* 11.1 (2016), pp. 31–50.
- [137] Astrid Vannoppen and Anne Gobin. "Estimating farm wheat yields from NDVI and meteorological data." In: *Agronomy* 11.5 (2021), p. 946.
- [138] AV Rozhkova and SE Rozhkov. "Artificial intelligence technologies in the agro-industrial complex: opportunities and threats." In: *IOP Conference Series: Earth and Environmental Science*. Vol. 981. 3. IOP Publishing. 2022, p. 032013.
- [139] Thomas Van Klompenburg, Ayalew Kassahun, and Cagatay Catal. "Crop yield prediction using machine learning: A systematic literature review." In: *Computers and Electronics in Agriculture* 177 (2020), p. 105709.
- [140] Deepak Vasisht et al. "{FarmBeats}: an {IoT} platform for {Data-Driven} agriculture." In: *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. 2017, pp. 515–529.
- [141] Kurt Benke and Bruce Tomkins. "Future food-production systems: vertical farming and controlled-environment agriculture." In: *Sustainability: Science, Practice and Policy* 13.1 (2017), pp. 13–26.
- [142] Steven W Van Ginkel, Thomas Igou, and Yongsheng Chen. "Energy, water and nutrient impacts of California-grown vegetables compared to controlled environmental agriculture systems in Atlanta, GA." In: *Resources, Conservation and Recycling* 122 (2017), pp. 319–325.
- [143] Dickson Despommier. "Farming up the city: The rise of urban vertical farms." In: *Trends in biotechnology* 31.7 (2013), pp. 388–389.
- [144] Dickson Despommier. "Vertical farms, building a viable indoor farming model for cities." In: *Field Actions Science Reports. The Journal of Field Actions* Special Issue 20 (2019), pp. 68–73.
- [145] Gerard K O'Neill. "The colonization of space." In: *Physics Today* 27.9 (1974), pp. 32–40.
- [146] Ahmad F Subahi and Kheir Eddine Bouazza. "An intelligent IoT-based system design for controlling and monitoring greenhouse temperature." In: *IEEE Access* 8 (2020), pp. 125488–125500.

- [147] Marc W van Iersel. "Optimizing LED lighting in controlled environment agriculture." In: *Light Emitting Diodes for Agriculture: Smart Lighting* (2017), pp. 59–80.
- [148] Lucas McCartney and Mark Lefsrud. "Protected agriculture in extreme environments: a review of controlled environment agriculture in tropical, arid, polar, and urban locations." In: *Applied engineering in agriculture* 34.2 (2018), pp. 455–473.
- [149] Redmond R Shamshiri et al. "Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture." In: (2018).
- [150] Muhammad Shoaib Farooq et al. "Internet of Things in Greenhouse Agriculture: A Survey on Enabling Technologies, Applications, and Protocols." In: *IEEE Access* 10 (2022), pp. 53374–53397.
- [151] Dafni Despoina Avgoustaki and George Xydis. "How energy innovation in indoor vertical farming can improve food security, sustainability, and food safety?" In: *Advances in food security and sustainability*. Vol. 5. Elsevier, 2020, pp. 1–51.
- [152] Bernard Ijesunor Akhigbe et al. "IoT technologies for livestock management: a review of present status, opportunities, and future trends." In: *Big data and cognitive computing* 5.1 (2021), p. 10.
- [153] Muhammad Shoaib Farooq et al. "A survey on the role of iot in agriculture for the implementation of smart livestock environment." In: *IEEE Access* 10 (2022), pp. 9483–9505.
- [154] Wataru Iwasaki, Nobutomo Morita, and Maria Portia Briones Nagata. "IoT sensors for smart livestock management." In: *Chemical, Gas, and Biosensors for Internet of Things and Related Applications*. Elsevier, 2019, pp. 207–221.
- [155] GS Karthick, M Sridhar, and PB Pankajavalli. "Internet of things in animal healthcare (IoT-AH): review of recent advancements in architecture, sensing technologies and real-time monitoring." In: *SN Computer Science* 1 (2020), pp. 1–16.
- [156] V Gokul and Sitaram Tadepalli. "Implementation of smart infrastructure and non-invasive wearable for real time tracking and early identification of diseases in cattle farming using IoT." In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE. 2017, pp. 469–476.
- [157] Zack Butler et al. "Virtual fences for controlling cows." In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*. Vol. 5. IEEE. 2004, pp. 4429–4436.

- [158] Yunhe Feng et al. "SocialCattle: IoT-based mastitis detection and control through social cattle behavior sensing in smart farms." In: *IEEE Internet of Things Journal* 9.12 (2021), pp. 10130–10138.
- [159] Ousmane Dieng et al. "A study on IoT solutions for preventing cattle rustling in african context." In: *ICC*. Vol. 17. 2017, pp. 1–13.
- [160] Meonghun Lee. "IoT livestock estrus monitoring system based on machine learning." In: *Asia-pacific Journal of Convergent Research Interchange* 4.3 (2018), pp. 119–128.
- [161] Santosh Kumar and Sanjay Kumar Singh. "Cattle recognition: A new frontier in visual animal biometrics research." In: *Proceedings of the national academy of sciences, India Section A: Physical Sciences* 90.4 (2020), pp. 689–708.
- [162] Marzieh Ronaghi and Mohammad Hossein Ronaghi. "Investigating the impact of economic, political, and social factors on augmented reality technology acceptance in agriculture (livestock farming) sector in a developing country." In: *Technology in Society* 67 (2021), p. 101739.
- [163] Mike O Ojo, Davide Adami, and Stefano Giordano. "Network performance evaluation of a LoRa-based IoT system for crop protection against ungulates." In: *2020 IEEE 25th international workshop on computer aided modeling and design of communication links and networks (CAMAD)*. IEEE. 2020, pp. 1–6.
- [164] Andrea Santangeli et al. "Integrating drone-borne thermal imaging with artificial intelligence to locate bird nests on agricultural land." In: *Scientific reports* 10.1 (2020), p. 10993.
- [165] Effie Papargyropoulou et al. "The food waste hierarchy as a framework for the management of food surplus and food waste." In: *Journal of cleaner production* 76 (2014), pp. 106–115.
- [166] Shobha Rathore et al. "Blockchain-based smart wheat supply chain model in Indian context." In: *Agri-Food 4.0*. Vol. 27. Emerald Publishing Limited, 2022, pp. 77–96.
- [167] Tarushi Wasson et al. "Integration of RFID and sensor in agriculture using IOT." In: *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. IEEE. 2017, pp. 217–222.
- [168] Matthew Edward, Kanisius Karyono, and H Meidia. "Smart fridge design using NodeMCU and home server based on Raspberry Pi 3." In: *2017 4th International Conference on New Media Studies (CONMEDIA)*. IEEE. 2017, pp. 148–151.

- [169] Bo Yan et al. "Information sharing in supply chain of agricultural products based on the Internet of Things." In: *Industrial Management & Data Systems* 116.7 (2016), pp. 1397–1416.
- [170] Sanjeev Yadav, Dixit Garg, and Sunil Luthra. "Analysing challenges for internet of things adoption in agriculture supply chain management." In: *International Journal of Industrial and Systems Engineering* 36.1 (2020), pp. 73–97.
- [171] Masanobu Fukuoka. "The natural way of farming." In: *Tokyo: Japan Publications* 17 (1985).
- [172] Rinku Singh and GS Singh. "Traditional agriculture: a climate-smart approach for sustainable food production." In: *Energy, Ecology and Environment* 2 (2017), pp. 296–316.
- [173] Heike Bach and Wolfram Mauser. "Sustainable agriculture and smart farming." In: *Earth observation open science and innovation* (2018), pp. 261–269.
- [174] Pankaj Agarwal et al. "Sustainable Smart-farming framework: smart farming." In: *Smart farming technologies for sustainable agricultural development*. IGI Global, 2019, pp. 147–173.
- [175] Jan Schoenke et al. "Gaia-AgStream: An Explainable AI Platform for Mining Complex Data Streams in Agriculture." In: *Smart and Sustainable Agriculture: First International Conference, SSA 2021, Virtual Event, June 21-22, 2021, Proceedings 1*. Springer, 2021, pp. 71–83.
- [176] Adrian Zărnescu et al. "Putting Internet-of-things at the service of sustainable agriculture. Case study: Sysagria." In: (2019).
- [177] Pradyumna K Tripathy et al. "MyGreen: An IoT-enabled smart greenhouse for sustainable agriculture." In: *IEEE Consumer Electronics Magazine* 10.4 (2021), pp. 57–62.
- [178] Redmond R Shamshiri et al. "Research and development in agricultural robotics: A perspective of digital farming." In: (2018).
- [179] Anish Paul Antony et al. "A review of practice and implementation of the internet of things (IoT) for smallholder agriculture." In: *Sustainability* 12.9 (2020), p. 3750.
- [180] Sakae Shibusawa. "Precision farming approaches for small scale farms." In: *IFAC Proceedings Volumes* 34.11 (2001), pp. 22–27.
- [181] Tamás Mizik. "Climate-smart agriculture on small-scale farms: A systematic literature review." In: *Agronomy* 11.6 (2021), p. 1096.

- [182] Anish Paul Antony, Daniel Sweeney, and Jennifer Lu. "Seeds of Silicon: Internet of Things for Smallholder Agriculture." In: (2019).
- [183] Ange Uwayisenga. "IoT-based system for automated floodwater detection and early warning in the East Africa Region: a case study of Arusha and Dar es salaam, Tanzania." PhD thesis. NM-AIST, 2021.
- [184] Shadi Al-Sarawi et al. "Internet of Things (IoT) communication protocols." In: *2017 8th International conference on information technology (ICIT)*. IEEE. 2017, pp. 685–690.
- [185] Farshad Firouzi, Bahar Farahani, and Alexander Marinšek. "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)." In: *Information Systems* 107 (2022), p. 101840.
- [186] Kyriakos Georgiou, Samuel Xavier-de-Souza, and Kerstin Eder. "The IoT energy challenge: A software perspective." In: *IEEE Embedded Systems Letters* 10.3 (2017), pp. 53–56.
- [187] url: <https://www.befc.global/>.
- [188] Teodora Sanislav et al. "Energy harvesting techniques for internet of things (IoT)." In: *IEEE Access* 9 (2021), pp. 39530–39549.
- [189] Shruti G Hegde Soumyalatha. "Study of IoT: understanding IoT architecture, applications, issues and challenges." In: *1st International Conference on Innovations in Computing & Net-working (ICICN16), CSE, RRCE. International Journal of Advanced Networking & Applications*. Vol. 478. 2016.
- [190] Wei Wang et al. "Knowledge representation in the internet of things: semantic modelling and its applications." In: *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije* 54.4 (2013), pp. 388–400.
- [191] Yuchen Yang et al. "A survey on security and privacy issues in Internet-of-Things." In: *IEEE Internet of things Journal* 4.5 (2017), pp. 1250–1258.
- [192] Sarah A Al-Qaseemi et al. "IoT architecture challenges and issues: Lack of standardization." In: *2016 Future technologies conference (FTC)*. IEEE. 2016, pp. 731–738.
- [193] Razvan Nicolescu et al. "Mapping the values of IoT." In: *Journal of Information Technology* 33 (2018), pp. 345–360.
- [194] Rushan Arshad et al. "Green IoT: An investigation on energy saving practices for 2020 and beyond." In: *Ieee Access* 5 (2017), pp. 15667–15681.

- [195] Spyros G Tzafestas. "Ethics and law in the internet of things world." In: *Smart cities* 1.1 (2018), pp. 98–120.
- [196] Mahmoud Abbasi, Mohammad Hossein Yaghmaee, and Fereshteh Rahnema. "Internet of Things in agriculture: A survey." In: *2019 3rd International Conference on Internet of Things and Applications (IoT)*. IEEE. 2019, pp. 1–12.
- [197] Nurzaman Ahmed, Debashis De, and Iftekhar Hussain. "Internet of Things (IoT) for smart precision agriculture and farming in rural areas." In: *IEEE Internet of Things Journal* 5.6 (2018), pp. 4890–4899.
- [198] Zhaoyun Zhang and Jingpeng Li. "A Review of Artificial Intelligence in Embedded Systems." In: *Micromachines* 14.5 (2023), p. 897.
- [199] Mustafa Kocakulak and Ismail Butun. "An overview of Wireless Sensor Networks towards internet of things." In: *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*. IEEE. 2017, pp. 1–6.
- [200] Naila Bouchemal, Sondes Kallel, and Nardjes Bouchemal. "A survey: WSN heterogeneous architecture platform for IoT." In: *Machine Learning for Networking: First International Conference, MLN 2018, Paris, France, November 27–29, 2018, Revised Selected Papers 1*. Springer. 2019, pp. 321–332.
- [201] SR Jino Ramson and D Jackuline Moni. "Applications of wireless sensor networks—A survey." In: *2017 international conference on innovations in electrical, electronics, instrumentation and media technology (ICEEIMT)*. IEEE. 2017, pp. 325–329.
- [202] Muhammad Asif et al. "Quality of service of routing protocols in wireless sensor networks: A review." In: *IEEE Access* 5 (2017), pp. 1846–1871.
- [203] Jolly Soparia and Nirav Bhatt. "A survey on comparative study of wireless sensor network topologies." In: *International Journal of Computer Applications* 87.1 (2014).
- [204] Azman Osman Lim and Youiti Kado. "Deployment and experimental verification for data communication in heterogeneous wireless networks." In: *2011 7th International Conference on Information Technology in Asia*. IEEE. 2011, pp. 1–7.
- [205] Ms Swati V Dhage, AN Thakre, SW Mohod, et al. "A review on scalability issue in wireless sensor networks." In: *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* 1 (2014), pp. 463–466.

- [206] Cheng-Chi Lee. "Security and privacy in wireless sensor networks: Advances and challenges." In: *Sensors* 20.3 (2020), p. 744.
- [207] Sushil Vilas Kalore and Prashant Rewagad. "A review on efficient routing techniques in wireless sensor networks." In: *2015 International Conference on Advances in Computer Engineering and Applications*. IEEE. 2015, pp. 803–807.
- [208] Kevin Kerliu et al. "Secure over-the-air firmware updates for sensor networks." In: *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*. IEEE. 2019, pp. 97–100.
- [209] Junaid Ahmed Khan, Hassaan Khaliq Qureshi, and Adnan Iqbal. "Energy management in wireless sensor networks: A survey." In: *Computers & Electrical Engineering* 41 (2015), pp. 159–176.
- [210] Feb. 1980. url: <https://www.iso.org/fr/ics/35.100/x/>.
- [211] Lydia Parziale et al. "TCP/IP tutorial and technical overview." In: (2006).
- [212] Praveen Kumar Donta et al. "Survey on recent advances in IoT application layer protocols and machine learning scope for research directions." In: *Digital Communications and Networks* 8.5 (2022), pp. 727–744.
- [213] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. "MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks." In: *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*. IEEE. 2008, pp. 791–798.
- [214] Zach Shelby, Klaus Hartke, and Carsten Bormann. *The constrained application protocol (CoAP)*. Tech. rep. 2014.
- [215] Steve Vinoski. "Advanced message queuing protocol." In: *IEEE Internet Computing* 10.6 (2006), pp. 87–89.
- [216] Bharati Wukkadada et al. "Comparison with HTTP and MQTT in Internet of Things (IoT)." In: *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE. 2018, pp. 249–253.
- [217] Nitin Naik. "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP." In: *2017 IEEE international systems engineering symposium (ISSE)*. IEEE. 2017, pp. 1–7.
- [218] Wentao Shang et al. "Challenges in IoT networking via TCP/IP architecture." In: *NDN Project 2* (2016).

- [219] Oladayo Bello, Sherali Zeadally, and Mohamad Badra. "Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT)." In: *Ad Hoc Networks* 57 (2017), pp. 52–62.
- [220] Zach Shelby and Carsten Bormann. *6LoWPAN: The wireless embedded Internet*. John Wiley & Sons, 2011.
- [221] Daquan Feng et al. "A survey of energy-efficient wireless communications." In: *IEEE Communications Surveys & Tutorials* 15.1 (2012), pp. 167–178.
- [222] Karan Nair et al. "Optimizing power consumption in iot based wireless sensor networks using Bluetooth Low Energy." In: *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*. IEEE. 2015, pp. 589–593.
- [223] Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. "IEEE 802.15. 4e: A survey." In: *Computer Communications* 88 (2016), pp. 1–24.
- [224] Klaus Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John wiley & sons, 2010.
- [225] Antonio Lazaro, Ramon Villarino, and David Girbau. "A survey of NFC sensors based on energy harvesting for IoT applications." In: *Sensors* 18.11 (2018), p. 3746.
- [226] Boris Bellalta et al. "Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges." In: *Computer Communications* 75 (2016), pp. 1–25.
- [227] Muneer Bani Yassein, Wail Mardini, and Ashwaq Khalil. "Smart homes automation using Z-wave protocol." In: *2016 International Conference on Engineering & MIS (ICEMIS)*. IEEE. 2016, pp. 1–6.
- [228] Mojtaba Vaezi et al. "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G." In: *IEEE Communications Surveys & Tutorials* 24.2 (2022), pp. 1117–1174.
- [229] Marco Centenaro et al. "A survey on technologies, standards and open challenges in satellite IoT." In: *IEEE Communications Surveys & Tutorials* 23.3 (2021), pp. 1693–1720.
- [230] Syed A Ahson and Mohammad Ilyas. *WiMAX: applications*. CRC press, 2018.
- [231] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. "Low power wide area networks: An overview." In: *ieee communications surveys & tutorials* 19.2 (2017), pp. 855–873.

- [232] ML Liya and D Arjun. "A survey of LPWAN technology in agricultural field." In: *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE. 2020, pp. 313–317.
- [233] Bharat S Chaudhari, Marco Zennaro, and Suresh Borkar. "LPWAN technologies: Emerging application characteristics, requirements, and design considerations." In: *Future Internet* 12.3 (2020), p. 46.
- [234] Dali Ismail, Mahbubur Rahman, and Abusayeed Saifullah. "Low-power wide-area networks: opportunities, challenges, and directions." In: *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*. 2018, pp. 1–6.
- [235] Alexandru Lavric, Adrian I Petrariu, and Valentin Popa. "Sigfox communication protocol: The new era of iot?" In: *2019 international conference on sensing and instrumentation in IoT Era (ISSI)*. IEEE. 2019, pp. 1–4.
- [236] Martin Bor, John Edward Vidler, and Utz Roedig. "LoRa for the Internet of Things." In: (2016).
- [237] Rapeepat Ratasuk et al. "NB-IoT system for M2M communication." In: *2016 IEEE wireless communications and networking conference*. IEEE. 2016, pp. 1–5.
- [238] Rapeepat Ratasuk et al. "Narrowband LTE-M system for M2M communication." In: *2014 IEEE 80th vehicular technology conference (VTC2014-Fall)*. IEEE. 2014, pp. 1–5.
- [239] Junxiao Dai and Masashi Sugano. "Low-cost sensor network for collecting real-time data for agriculture by combining energy harvesting and LPWA technology." In: *2019 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE. 2019, pp. 1–4.
- [240] Ana Pejković, Josip Spišić, and Matko Zrnić. "LPWAN for use in agriculture." In: *PhD Forum–Book of Abstracts*. 2021, p. 11.
- [241] Laura Joris et al. "An autonomous sigfox wireless sensor node for environmental monitoring." In: *IEEE Sensors Letters* 3.7 (2019), pp. 01–04.
- [242] Feb. 2023. url: <https://sencrop.com>.
- [243] Alvaro Llaría et al. "Geolocation and monitoring platform for extensive farming in mountain pastures." In: *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2015, pp. 2420–2425.

- [244] FigaroTech. *Sigfox, fin de partie pour une ex-star de la French Tech*. Jan. 2022. url: <https://www.lefigaro.fr/secteur/high-tech/sigfox-la-fin-du-reve-pour-la-0g-20220127>.
- [245] Giovanni Valecce et al. "NB-IoT for smart agriculture: Experiments from the field." In: *2020 7th international conference on control, Decision and Information Technologies (CoDIT)*. Vol. 1. IEEE. 2020, pp. 71–75.
- [246] Jiajian Yang, Zhenfeng Xu, and Qiang Sun. "Research and design of greenhouse environment monitoring system based on NB-IoT." In: *2021 33rd Chinese Control and Decision Conference (CCDC)*. IEEE. 2021, pp. 5641–5646.
- [247] Haitao Ye, Yongjie Yang, and Linyu Zhu. "A wireless network detection and control system for intelligent agricultural greenhouses based on NB-IOT technology." In: *Journal of Physics: Conference Series*. Vol. 1738. 1. IOP Publishing. 2021, p. 012058.
- [248] Joao Cardoso, André Glória, and Pedro Sebastiao. "A Methodology for Sustainable Farming Irrigation using WSN, NB-IoT and Machine Learning." In: *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE. 2020, pp. 1–6.
- [249] Juan Huan et al. "Design of water quality monitoring system for aquaculture ponds based on NB-IoT." In: *Aquacultural Engineering* 90 (2020), p. 102088.
- [250] Sikandar Zulqarnain Khan, Yannick Le Moullec, and Muhammad Mahtab Alam. "An nb-iot-based edge-of-things framework for energy-efficient image transfer." In: *Sensors* 21.17 (2021), p. 5929.
- [251] Ivan Hu and Zihuai Lin. "Design of an embedded system with on-demand image capturing and transmission for remote agricultural monitoring." In: *arXiv preprint arXiv:2205.12441* (2022).
- [252] Jose A Brenes and Gabriela Marín-Raventós. "When one wireless technology is not enough: A network architecture for precision agriculture using LoRa, Wi-Fi, and LTE." In: *Intelligent Sustainable Systems: Selected Papers of WorldS4 2021, Volume 1*. Springer. 2022, pp. 103–112.
- [253] Danco Davcev et al. "IoT agriculture system based on LoRaWAN." In: *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*. IEEE. 2018, pp. 1–4.

- [254] Jehangir Arshad et al. "Implementation of a LoRaWAN based smart agriculture decision support system for optimum crop yield." In: *Sustainability* 14.2 (2022), p. 827.
- [255] Badreddine Miles et al. "A study of LoRaWAN protocol performance for IoT applications in smart agriculture." In: *Computer Communications* 164 (2020), pp. 148–157.
- [256] Ritesh Kumar Singh et al. "Leveraging LoRaWAN technology for precision agriculture in greenhouses." In: *Sensors* 20.7 (2020), p. 1827.
- [257] Pongsarun Boonyopakorn and Thanamate Thongna. "Environment monitoring system through LoRaWAN for smart agriculture." In: *2020-5th International Conference on Information Technology (InCIT)*. IEEE. 2020, pp. 12–16.
- [258] Esteban Sánchez Hernández et al. "LoRaWAN applied to agriculture: A use case for automated irrigation systems." In: *International Conference on Disruptive Technologies, Tech Ethics and Artificial Intelligence*. Springer. 2021, pp. 308–316.
- [259] Mookeun Ji et al. "Lora-based visual monitoring scheme for agriculture iot." In: *2019 IEEE sensors applications symposium (SAS)*. IEEE. 2019, pp. 1–6.
- [260] C Joshitha et al. "Lorawan based cattle monitoring smart system." In: *2021 7th International Conference on Electrical Energy Systems (ICEES)*. IEEE. 2021, pp. 548–552.
- [261] Mohammad Al mojamed. "On the use of LoRaWAN for mobile Internet of Things: The impact of mobility." In: *Applied System Innovation* 5.1 (2021), p. 5.
- [262] Chaojie Gu, Linshan Jiang, and Rui Tan. "Lora-based localization: Opportunities and challenges." In: *arXiv preprint arXiv:1812.11481* (2018).
- [263] Weisong Shi et al. "Edge computing: Vision and challenges." In: *IEEE internet of things journal* 3.5 (2016), pp. 637–646.
- [264] Flavio Bonomi et al. "Fog computing and its role in the internet of things." In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012, pp. 13–16.
- [265] Dan C Marinescu. *Cloud computing: theory and practice*. Morgan Kaufmann, 2022.
- [266] Mohammed Laroui et al. "Edge and fog computing for IoT: A survey on current research activities & future directions." In: *Computer Communications* 180 (2021), pp. 210–231.

- [267] Toru Baji. "Evolution of the GPU Device widely used in AI and Massive Parallel Processing." In: *2018 IEEE 2nd Electron devices technology and manufacturing conference (EDTM)*. IEEE. 2018, pp. 7–9.
- [268] Stephen Cass. "Nvidia makes it easy to embed AI: The Jetson nano packs a lot of machine-learning power into DIY projects-[Hands on]." In: *IEEE Spectrum* 57.7 (2020), pp. 14–16.
- [269] Lai Jia Shyan, TH Lim, and Dk Norhafizah Pg Hj Muhammad. "Real time road traffic sign detection and recognition systems using Convolution Neural Network on a GPU platform." In: *2022 International Conference on Digital Transformation and Intelligence (ICDI)*. IEEE. 2022, pp. 236–240.
- [270] Wai-Kong Lee et al. "Efficient implementation of lightweight hash functions on gpu and quantum computers for iot applications." In: *IEEE Access* 10 (2022), pp. 59661–59674.
- [271] Yuan Lei et al. "Low power AI ASIC design for portable edge computing." In: *2020 IEEE 15th International Conference on Solid-State & Integrated Circuit Technology (ICSICT)*. IEEE. 2020, pp. 1–4.
- [272] "Accelerate AI development with Google Cloud TPUs." In: ().
- [273] Shiho Kim and Ganesh Chandra Deka. *Hardware accelerator systems for artificial intelligence and machine learning*. Academic Press, 2021.
- [274] Zane Tsai. "Embedded Hardware for Processing AI at the Edge: GPU, VPU, FPGA, and ASIC Explained." In: (2021).
- [275] Zhengjie Li et al. "A survey of FPGA design for AI era." In: *Journal of Semiconductors* 41.2 (2020), p. 021402.
- [276] Kah Phooi Seng, Paik Jen Lee, and Li Minn Ang. "Embedded intelligence on FPGA: Survey, applications and challenges." In: *Electronics* 10.8 (2021), p. 895.
- [277] Miruna A Clinciu and Helen F Hastie. "A survey of explainable AI terminology." In: *1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence 2019*. Association for Computational Linguistics. 2019, pp. 8–13.
- [278] Yu Zhang et al. "A survey on neural network interpretability." In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742.
- [279] Sebastian Kujawa and Gniewko Niedbała. *Artificial neural networks in agriculture*. 2021.

- [280] Lachit Dutta and Swapna Bharali. "Tinyml meets iot: A comprehensive survey." In: *Internet of Things* 16 (2021), p. 100461.
- [281] Rukshan Pramoditha. *Overview of a neural network's learning process*. Feb. 2022. url: <https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa>.
- [282] Keras Team. *Keras Documentation: Losses*. url: <https://keras.io/api/losses/>.
- [283] Juan Terven and Diana Cordova-Esparza. *A Comprehensive Review of YOLO: From YOLOv1 and Beyond*. 2023. arXiv: 2304.00501 [cs.CV].
- [284] Lei Deng et al. "Model compression and hardware acceleration for neural networks: A comprehensive survey." In: *Proceedings of the IEEE* 108.4 (2020), pp. 485–532.
- [285] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." In: *arXiv preprint arXiv:1704.04861* (2017).
- [286] Muhammad Sabih, Frank Hannig, and Juergen Teich. "Utilizing explainable AI for quantization and pruning of deep neural networks." In: *arXiv preprint arXiv:2008.09072* (2020).
- [287] Tailin Liang et al. "Pruning and quantization for deep neural network acceleration: A survey." In: *Neurocomputing* 461 (2021), pp. 370–403.
- [288] Shuai Zhu et al. "On-device Training: A First Overview on Existing Systems." In: *arXiv preprint arXiv:2212.00824* (2022).
- [289] Haoyu Ren, Darko Anicic, and Thomas A Runkler. "Tinyol: Tinyml with online-learning on microcontrollers." In: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2021, pp. 1–8.
- [290] Anas Osman et al. "Tinyml platforms benchmarking." In: *International Conference on Applications in Electronics Pervading Industry, Environment and Society*. Springer. 2021, pp. 139–148.
- [291] Liliana Andrade, Adrien Prost-Boucle, and Frédéric Pétrot. "Overview of the state of the art in embedded machine learning." In: *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2018, pp. 1033–1038.
- [292] Youssef Abadade et al. "A Comprehensive Survey on TinyML." In: *IEEE Access* (2023).

- [293] Hui Han and Julien Siebert. "TinyML: A systematic review and synthesis of existing research." In: *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*. IEEE. 2022, pp. 269–274.
- [294] Bárbara Costa, Octavian Postolache, and John Araujo. "From cloud AI to embedded AI in cardiac healthcare." In: *2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE. 2023, pp. 1–6.
- [295] H Gokul et al. "Gait recovery system for parkinson's disease using machine learning on embedded platforms." In: *2020 IEEE International Systems Conference (SysCon)*. IEEE. 2020, pp. 1–8.
- [296] Vitor M Oliveira and António HJ Moreira. "Edge AI System Using a Thermal Camera for Industrial Anomaly Detection." In: *International Summit Smart City 360°*. Springer, 2021, pp. 172–187.
- [297] Mansoureh Lord and Adam Kaplan. "Mechanical Anomaly Detection on an Embedded Microcontroller." In: *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2021, pp. 562–568.
- [298] Latifa M Mrisho et al. "Accuracy of a smartphone-based object detection model, PlantVillage Nuru, in identifying the foliar symptoms of the viral diseases of cassava—CMD and CBSD." In: *Frontiers in plant science* 11 (2020), p. 590889.
- [299] Paula Ramos-Giraldo et al. "Low-cost smart camera system for water stress detection in crops." In: *2020 IEEE SENSORS*. IEEE. 2020, pp. 1–4.
- [300] Laura Falaschetti et al. "A low-cost, low-power and real-time image detector for grape leaf esca disease based on a compressed CNN." In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 11.3 (2021), pp. 468–481.
- [301] Fabrizio De Vita et al. "Quantitative analysis of deep leaf: a plant disease detector on the smart edge." In: *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE. 2020, pp. 49–56.
- [302] Vivek Gutti and R Karthi. "Real Time Classification of Fruits and Vegetables Deployed on Low Power Embedded Devices Using Tiny ML." In: *International Conference on Image Processing and Capsule Networks*. Springer. 2022, pp. 347–359.

- [303] Sharfuddin Waseem Mohammed et al. "Smart Water Resource Management by Analyzing the Soil Structure and Moisture Using Deep Learning." In: *IOT with Smart Systems: Proceedings of ICTIS 2022, Volume 2*. Springer, 2022, pp. 709–719.
- [304] Claudia Bruno et al. "Embedded artificial intelligence approach for gas recognition in smart agriculture applications using low cost mox gas sensors." In: *2021 Smart Systems Integration (SSI)*. IEEE. 2021, pp. 1–5.
- [305] Reza Arablouei et al. "Animal behavior classification via deep learning on embedded systems." In: *Computers and Electronics in Agriculture* 207 (2023), p. 107707.
- [306] HR Sabbella et al. "An always-on tinyml acoustic classifier for ecological applications." In: *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2022, pp. 2393–2396.
- [307] John K Debenham. *Knowledge systems design*. Prentice-Hall, Inc., 1989.
- [308] Rudi Studer, V Richard Benjamins, and Dieter Fensel. "Knowledge engineering: Principles and methods." In: *Data & knowledge engineering* 25.1-2 (1998), pp. 161–197.
- [309] Fred Nickols. "The tacit and explicit nature of knowledge: The knowledge in knowledge management." In: *The knowledge management yearbook 2000-2001*. Routledge, 2013, pp. 12–21.
- [310] Simon L Kendal and Malcolm Creen. *An introduction to knowledge engineering*. Springer, 2007.
- [311] Peter Jackson. "Introduction to expert systems." In: (1986).
- [312] Nigam Shah and Mark Musen. "Ontologies for formal representation of biological systems." In: *Handbook on ontologies*. Springer, 2009, pp. 445–461.
- [313] url: https://www.w3.org/2001/sw/wiki/Main_Page.
- [314] Ahlem Rhayem, Mohamed Ben Ahmed Mhiri, and Faiez Gargouri. "Semantic web technologies for the internet of things: Systematic literature review." In: *Internet of Things* 11 (2020), p. 100206.
- [315] I Szilagyi and P Wira. "Ontologies and Semantic Web for the Internet of Things-a survey, IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society." In: *Florence 2016* (2016), pp. 6949–6954.
- [316] url: <https://www.w3.org/TR/vocab-ssn/>.

- [317] Dave Raggett. "The web of things: Challenges and opportunities." In: *Computer* 48.5 (2015), pp. 26–32.
- [318] Rimel Bendadouche et al. "Extension of the semantic sensor network ontology for wireless sensor networks: The stimulus-WSNnode-communication pattern." In: *5th International Workshop on Semantic Sensor Networks in conjunction with the 11th International Semantic Web Conference (ISWC)*. 2012, 16–p.
- [319] Heiko Müller et al. "From RESTful to SPARQL: A Case Study on Generating Semantic Sensor Data." In: *SSN@ ISWC*. 2013, pp. 51–66.
- [320] Brett Drury et al. "A survey of semantic web technology for agriculture." In: *Information Processing in Agriculture* 6.4 (2019), pp. 487–501.
- [321] Bikram Pratim Bhuyan, Ravi Tomar, and Amar Ramdane Cherif. "A Systematic Review of Knowledge Representation Techniques in Smart Agriculture (Urban)." In: *Sustainability* 14.22 (2022), p. 15249.
- [322] url: <https://www.fao.org/agrovoc/fr>.
- [323] Salvador Sánchez-Alonso and Miguel-Angel Sicilia. "Using an AGROVOC-based ontology for the description of learning resources on organic agriculture." In: *Metadata and Semantics*. Springer, 2009, pp. 481–492.
- [324] Clement Jonquet et al. "AgroPortal: A vocabulary and ontology repository for agronomy." In: *Computers and Electronics in Agriculture* 144 (2018), pp. 126–143.
- [325] url: <https://cropontology.org/>.
- [326] Laurel Cooper et al. "The Planteome database: an integrated resource for reference ontologies, plant genomics and phenomics." In: *Nucleic acids research* 46.D1 (2018), pp. D1168–D1180.
- [327] Valeria Pesce, Ajit Maru, and Johannes Keizer. "The CIARD RING, an infrastructure for interoperability of agricultural research information services." In: (2010).
- [328] url: <https://vest.agrisemantics.org/>.
- [329] Miguel Ángel Rodríguez-García and Francisco García-Sánchez. "CropPestO: An Ontology Model for Identifying and Managing Plant Pests and Diseases." In: *International Conference on Technologies and Innovation*. Springer. 2020, pp. 18–29.

- [330] Salvador Sánchez-Alonso et al. "Engineering an ontology on organic agriculture and agroecology: the case of the Organic. Edunet project." In: *Proc. of the World Conference on Agricultural Information and IT (IAALD AFITA WCCA 2008), Tokyo, Japan. 2008*, pp. 24–27.
- [331] Udsanee Pakdeetrakulwong and Kairung Hengpraprom. "An Ontology-based Knowledge Management for Organic Agriculture and Good Agricultural Practices: A Case Study of Nakhon Pathom Province, Thailand." In: *Interdisciplinary Research Review* 13.4 (2018), pp. 26–34.
- [332] Vincent Soullignac et al. "GECO, the French web-based application for knowledge management in agroecology." In: *Computers and Electronics in Agriculture* 162 (2019), pp. 1050–1056.
- [333] Sylvie Pouteau. "The plantness of the plant: how to address crop ontology in organic agriculture?" In: (2021).
- [334] Stephanie Domptail, Jennifer Hirsch, and Chukwuma Ume. "Agroecology as an ontology to guide agricultural and food systems?" In: (2021).
- [335] Jean Vincent Fonou-Dombeu et al. "OntoCSA: A Climate-Smart Agriculture Ontology." In: *International Journal of Agricultural and Environmental Information Systems (IJAEIS)* 12.4 (2021), pp. 1–20.
- [336] Nadia Naidoo et al. "Modelling Climate Smart Agriculture with Ontology." In: *2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE. 2021, pp. 1–9.
- [337] Adebayo Adewumi Abayomi-Alli et al. "An ontology-based information extraction system for organic farming." In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 17.2 (2021), pp. 79–99.
- [338] Kushala Vijaya Kumar Mummigatti, Supriya Maganahalli Chandramouli, and Divakar Harohally Ramachandra. "Deep Neural Network System Using Ontology to Recommend Organic Fertilizers for a Sustainable Agriculture." In: *Ingénierie des Systèmes d'Information* 28.2 (2023).
- [339] Muhammad Shoaib Farooq, Shamyla Riaz, and Atif Alvi. "Web of Things and Trends in Agriculture: A Systematic Literature Review." In: *arXiv preprint arXiv:2306.09079* (2023).

- [340] Siquan Hu et al. "AgOnt: ontology for agriculture internet of things." In: *Computer and Computing Technologies in Agriculture IV: 4th IFIP TC 12 Conference, CCTA 2010, Nanchang, China, October 22-25, 2010, Selected Papers, Part I 4*. Springer. 2011, pp. 131–137.
- [341] Andreas Kamilaris et al. "Agri-IoT: A semantic framework for Internet of Things-enabled smart farming applications." In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE. 2016, pp. 442–447.
- [342] Ganesh Ramanathan et al. "Semantic Knowledge for Autonomous Smart Farming." In: *IFAC-PapersOnLine* 55.32 (2022), pp. 217–222.
- [343] Moez Altayeb, Marco Zennaro, and Marcelo Rovai. "Classifying mosquito wingbeat sound using TinyML." In: *Proceedings of the 2022 ACM Conference on Information Technology for Social Good*. 2022, pp. 132–137.
- [344] Hoan Nguyen Duc et al. "Research and Implement Embedded Artificial Intelligence in Low-Power Water Meter Reading Device." In: *2021 International Conference on Advanced Technologies for Communications (ATC)*. IEEE. 2021, pp. 119–124.
- [345] Teoh Ji Sheng et al. "An internet of things based smart waste management system using LoRa and tensorflow deep learning model." In: *IEEE Access* 8 (2020), pp. 148793–148811.
- [346] Jean-Paul Bardyn et al. "IoT: The era of LPWAN is starting now." In: *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*. IEEE. 2016, pp. 25–30.
- [347] Aug. 2023. url: https://loro-alliance.org/?utm_id=France.
- [348] Brecht Reynders and Sofie Pollin. "Chirp spread spectrum as a modulation technique for long range communication." In: *2016 Symposium on Communications and Vehicular Technologies (SCVT)*. IEEE. 2016, pp. 1–5.
- [349] Norman Abramson. "The ALOHA system: Another alternative for computer communications." In: *Proceedings of the November 17-19, 1970, fall joint computer conference*. 1970, pp. 281–285.
- [350] Sjaak Wolfert et al. "Big data in smart farming—a review." In: *Agricultural systems* 153 (2017), pp. 69–80.
- [351] Akram H Jebiril et al. "Overcoming limitations of LoRa physical layer in image transmission." In: *Sensors* 18.10 (2018), p. 3257.

- [352] Chinc-Chuan Wei et al. "The Development of LoRa Image Transmission Based on Time Division Multiplexing." In: *2020 International Symposium on Computer, Consumer and Control (IS3C)*. IEEE. 2020, pp. 323–326.
- [353] Ching-Chung Wei, Shu-Ting Chen, and Pei-Yi Su. "Image transmission using LoRa technology with various spreading factors." In: *2019 2nd World Symposium on Communication Engineering (WSCE)*. IEEE. 2019, pp. 48–52.
- [354] Ching-Chuan Wei, Pei-Yi Su, and Shu-Ting Chen. "Comparison of the LoRa image transmission efficiency based on different encoding methods." In: *International Journal of Information and Electronics Engineering* 10.1 (2020), pp. 1–4.
- [355] Kevin Guerra et al. "A low-rate encoder for image transmission using LoRa communication modules." In: *International Journal of Information Technology* 15.2 (2023), pp. 1069–1079.
- [356] Ada Fort, Giacomo Peruzzi, and Alessandro Pozzebon. "Quasi-real time remote video surveillance unit for lorawan-based image transmission." In: *2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT)*. IEEE. 2021, pp. 588–593.
- [357] John YQ Zhang et al. "LoRaWAN-based Camera with (CIRA) Compression and Image Recovery Algorithm." In: *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*. IEEE. 2021, pp. 136–141.
- [358] Congduc Pham. "Low-cost, low-power and long-range image sensor for visual surveillance." In: *Proceedings of the 2nd Workshop on Experiences in the Design and Implementation of Smart Objects*. 2016, pp. 35–40.
- [359] Congduc Pham. "Robust CSMA for long-range LoRa transmissions with image sensing devices." In: *2018 Wireless Days (WD)*. IEEE. 2018, pp. 116–122.
- [360] L. Kleinrock and F. Tobagi. "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics." In: *IEEE Transactions on Communications* 23.12 (1975), pp. 1400–1416. doi: [10.1109/TCOM.1975.1092768](https://doi.org/10.1109/TCOM.1975.1092768).
- [361] Tonghao Chen, Derek Eager, and Dwight Makaroff. "Efficient image transmission using lora technology in agricultural monitoring iot systems." In: *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2019, pp. 937–944.

- [362] Konstantinos Arakadakis et al. "Firmware Over-the-Air Programming Techniques for IoT Networks - A Survey." In: *ACM Comput. Surv.* 54.9 (Oct. 2021). issn: 0360-0300. doi: [10.1145/3472292](https://doi.org/10.1145/3472292). url: <https://doi.org/10.1145/3472292>.
- [363] Saad El Jaouhari and Eric Bouvet. "Secure firmware Over-The-Air updates for IoT: Survey, challenges, and discussions." In: *Internet of Things* 18 (2022), p. 100508.
- [364] Meriem Bettayeb, Qassim Nasir, and Manar Abu Talib. "Firmware update attacks and security for IoT devices: Survey." In: *Proceedings of the ArabWIC 6th Annual International Conference Research Track*. 2019, pp. 1–6.
- [365] LoRa Alliance. "Fuota process summary technical recommendation." In: *TR002 v1. 0.0. [(accessed on 27 October 2020)]* (2019), pp. 2019–04.
- [366] Nicolas Sornin. "LoRaWAN Firmware Updates OTA." In: *LoRa for developers* (2020).
- [367] Khaled Abdelfadeel et al. "How to make firmware updates over lorawan possible." In: *2020 IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2020, pp. 16–25.
- [368] Derek Heeger et al. "Secure LoRa firmware update with adaptive data rate techniques." In: *Sensors* 21.7 (2021), p. 2384.
- [369] Tim Hess, David Bol, and Ramin Sadre. "Ultra-low-power over-the-air-update in secure lorawan networks." In: *Ecole polytechnique de Louvain, Université catholique de Louvain* (2020).
- [370] url: <https://docs.arduino.cc/hardware/portenta-h7>.
- [371] url: <https://docs.arduino.cc/resources/datasheets/ASX00026-datasheet.pdf>.
- [372] url: <https://www.st.com/en/evaluation-tools/p-nucleo-lrwan2.html#overview>.
- [373] url: https://www.uctronics.com/download/cam_module/OV2640DS.pdf.
- [374] url: https://www.espressif.com/en/news/ESP32_CAM.
- [375] url: <https://www.hoperf.com/modules/lora/RFM95.html>.
- [376] url: <https://www.st.com/en/ecosystems/stm32cube.html>.
- [377] url: <https://docs.platformio.org/en/latest/integration/ide/vscode.html>.

- [378] url: <https://www.thethingsnetwork.org/>.
- [379] url: <https://www.chirpstack.io/>.
- [380] url: <https://www.freertos.org/>.
- [381] url: <https://os.mbed.com/mbed-os/>.
- [382] url: edgeimpulse.com/.
- [383] url: <https://omnetpp.org/>.
- [384] url: <https://flora.aalto.fi/>.
- [385] Sadiq Idris, Thenuka Karunathilake, and Anna Förster. "Survey and comparative study of LoRa-enabled simulators for internet of things and wireless sensor networks." In: *Sensors* 22.15 (2022), p. 5546.
- [386] Chris Gorgolewski. *Fruit recognition*. Feb. 2020. url: <https://www.kaggle.com/datasets/chrisfilo/fruit-recognition?select=muskmelon>.
- [387] Kelcey Moreau. "FOMO: Object detection for constrained devices." In: (2021).
- [388] V Aadithya et al. "Comparative Study Between MobilNet Face-Mask Detector and YOLOv3 Face-Mask Detector." In: *Sustainable Communication Networks and Application: Proceedings of ICSCN 2021*. Springer, 2022, pp. 801–809.
- [389] Nicolas Chollet. "Portenta Fragola software." In: (2022). url: https://github.com/NicolasChollet51/Portenta_Fragola.
- [390] Amar Ramdane-Cherif Nicolas Chollet Naila Bouchemal. "Embedded AI and Computation Offloading for 6G Green Communication." In: *6gNet 2023* (2023). url: <https://6g-conference.dnac.org/>.
- [391] url: <https://github.com/chirpstack/chirpstack-fuota-server>.
- [392] Chollet Nicolas, Bouchemal Naila, and Ramdane-Cherif Amar. "Energy efficient Firmware Over The Air Update for TinyML models in LoRaWAN agricultural networks." In: *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE. 2022, pp. 21–27.
- [393] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.
- [394] url: <https://www.w3.org/TR/owl2-primer/>.

- [395] Franz Baader, Ian Horrocks, and Ulrike Sattler. "Description logics." In: *Foundations of Artificial Intelligence* 3 (2008), pp. 135–179.
- [396] Michael Debellis. "A practical guide to building OWL ontologies using Protégé 5.5 and plugins." In: URL: https://www.researchgate.net/publication/351037551_A_Practical_Guide_to_Building_OWL_Ontologies_Using_Protege : 11.05. 2023) (2021).
- [397] Mark A Musen. "The protégé project: a look back and a look forward." In: *AI matters* 1.4 (2015), pp. 4–12.
- [398] Evren Sirin et al. "Pellet: A practical owl-dl reasoner." In: *Journal of Web Semantics* 5.2 (2007), pp. 51–53.
- [399] Agroportal. *Agroportal/fairness: This project is a fairness assessment tool for ontologies, vocabularies and semantic resources*. url: <https://github.com/agroportal/fairness#-ofaire-ontology-fairness-evaluator>.
- [400] Emna Amdouni and Clement Jonquet. "FAIR or FAIRer? An integrated quantitative FAIRness assessment grid for semantic resources and ontologies." In: *Research Conference on Metadata and Semantics Research*. Springer. 2021, pp. 67–80.