



الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
المدرسة الوطنية العليا للتكنولوجيات المتقدمة
National Higher School of Advanced Technologies



Department of Electrical Engineering and Industrial Computing

End of Study Project to Obtain the Diploma of
Engineering

Field
Telecommunication

Speciality
Telecommunications Systems and Networks

Subject
**Synergizing IoT and AI for Sustainable
Greenhouse Management**

Realized by
ZEGHDAR Seif El Islam, SOUIKI Mohamed Chakib

Members of The Jury:

President	Ms. M. BOUTERFAS
Promoter	Ms. L. ABBAD
Promoter	Ms. K. LAKHDARI
Examiner	Mr. E. KHOUMERI
Examiner	Mr. A. HENTOUT

Algiers, the 20 / 06 /2024

Academic year 2023 – 2024

Dedication

“

This work stands as the culmination of countless sacrifices and arduous efforts. With heartfelt pride and deep gratitude, we dedicate this endeavor to all who have been our inspiration and support.

To our fathers, our mothers, and our entire families, whose unwavering love and encouragement have been our pillars of strength. To all our friends, whose support and assistance amidst challenges have been invaluable.

We extend our sincere appreciation to the faculty and staff of National Higher School of Advanced Technologies for their guidance and knowledge. Above all, we are grateful to our Almighty God, whose blessings of strength, courage, patience, wisdom, time, and guidance have illuminated our path towards the realization of this work.

”

Acknowledgment

Above all, we extend our deepest gratitude to Almighty God for blessing us with the courage, resilience, and patience that sustained us throughout our academic journey.

We wish to sincerely acknowledge and thank all those who have contributed to the completion of this thesis. Firstly, we are indebted to our supervisors, **Mrs. ABBAD Leila** and **Mrs. LAKHDARI Kheira**, for their invaluable guidance, unwavering patience, and continuous availability throughout this project.

A heartfelt thank you goes to all the members of the jury for their willingness to meticulously examine and evaluate our work. We also express our appreciation to all our teachers whose dedicated efforts have enriched our education over the years.

Our friends and families have been a constant source of support and motivation throughout the completion of this thesis. Their encouragement and belief in us have been instrumental in overcoming challenges and achieving our goals.

We acknowledge and appreciate the contributions of all individuals who have directly or indirectly played a role in the realization of this work. Your collective support has been integral to our success, and for that, we are truly grateful.

ملخص

تستكشف هذه الرسالة التأثير المتبادل بين إنترنت الأشياء (IoT) والذكاء الاصطناعي (AI) في ثورة الزراعة في البيوت الزراعية من أجل الاستدامة. نحلل التحديات التي تواجهها نمو السكان العالمي وتغير المناخ، مع التركيز على الحاجة إلى حلول مبتكرة. تعتمد منهجيتنا على إنترنت الأشياء لتوصيل الأجهزة الاستشعارية واكتساب البيانات، بينما تتيح خوارزميات الذكاء الاصطناعي اتخاذ القرارات الدقيقة والتحليلات التنبؤية. تفصل الرسالة عملية التنفيذ، بما في ذلك اختيار المعدات وبروتوكولات الاتصال وتطوير نماذج AI، مما يؤدي في النهاية إلى تطوير لوحة تحكم ويب ذكية سهلة الاستخدام لرصد الوقت الفعلي. من خلال التقييم التجريبي، نثبت فعالية حلنا في تعزيز الإنتاجية الزراعية والاستدامة.

الكلمات الرئيسية: الذكاء الاصطناعي، إنترنت الأشياء، البيوت الزراعية، الزراعة، الأجهزة الاستشعارية، المحركات، MQTT، سلاسل الزمن.

Abstract

This thesis explores the synergy between Internet of Things (IoT) and Artificial Intelligence (AI) in revolutionizing greenhouse agriculture for sustainability. We analyze the challenges posed by global population growth and climate change, emphasizing the need for innovative solutions. Our approach leverages IoT for sensor connectivity and data acquisition, while AI algorithms enable precise decision-making and predictive analytics. The thesis details the implementation process, including equipment selection, and communication protocols and AI model development, culminating in the development of a user-friendly smart web application dashboard for real-time monitoring. Through empirical evaluation, we demonstrate the efficacy of our solution in enhancing agricultural productivity and sustainability.

Key words: Artificial Intelligence, Internet of Things, Greenhouse, Agriculture, Sensor, Actuator, MQTT, Time series.

Résumé

Cette thèse explore la synergie entre l'Internet des objets (IoT) et l'intelligence artificielle (AI) dans la révolution de l'agriculture sous serre pour la durabilité. Nous analysons les défis posés par la croissance démographique mondiale et le changement climatique, en mettant l'accent sur la nécessité de solutions innovantes. Notre approche exploite l'IoT pour la connectivité des capteurs et l'acquisition de données, tandis que les algorithmes d'IA permettent une prise de décision précise et des analyses prédictives. La thèse détaille le processus de mise en œuvre, y compris la sélection de l'équipement, les protocoles de communication et le développement de modèles d'IA, aboutissant au développement d'un tableau de bord d'application web intelligente convivial pour la surveillance en temps réel. Grâce à une évaluation empirique, nous démontrons l'efficacité de notre solution dans l'amélioration de la productivité agricole et de la durabilité.

Mots clés: Intelligence artificielle, Internet des Objets, serre, agriculture, capteur, actionneur, MQTT, séries temporelles.

Table of Contents

List of Figures	1
List of Tables	3
List of Abbreviations.....	4
General Introduction	6
CHAPTER 1. IoT and Agriculture.....	8
1.1. Introduction.....	9
1.2. Understanding IoT	9
1.2.1. IoT Architecture Overview	10
1.3. IoT Today: The Tipping Point.....	11
1.4. IoT Applications in General	11
1.5. IoT in Agriculture.....	13
1.6. IoT in Greenhouses.....	14
1.6.1. Related Work	14
1.7. Advantages and Disadvantages of IoT in Agriculture	17
1.7.1. Advantages.....	17
1.7.2. Disadvantages.....	17
1.8. IoT Communication Protocols	18
1.8.1. MQTT Definition	19
1.8.2. MQTT Architecture	19
1.8.3. MQTT Message Format.....	19
1.9. Conclusion.....	20
CHAPTER 2. AI and Agriculture	21
2.1. Introduction.....	22
2.2. Artificial Intelligence	22
2.2.1. Theoretical Foundations of AI	22
2.3. Key Areas and Applications of AI	25
2.4. Importance of AI in Modern Technology and Industry.....	26
2.5. AI and Agriculture.....	27
2.5.1 Nature of Data Utilized in Agricultural AI:.....	27
2.5.2 Application of AI Technologies in Agriculture	28
2.5.3 Challenges in AI Adoption for Agriculture.....	30
2.6. Time Series.....	31
2.6.1 Definition.....	31

Table of Contents

2.6.2	Components.....	31
2.7	Time Series in Agriculture.....	32
2.8	Time Series in Greenhouses.....	32
2.9	Time Series Forecasting.....	33
2.9.1	Forecasting Methods.....	33
2.9.2	Forecasting Models	34
CHAPTER 3. GreenSense AI: Architecture, Workflow, and Tools		42
3.1.	Introduction.....	43
3.2	GreenSense AI	43
3.2.1	The Meaning Behind "GreenSense AI"	43
3.2.2	Solution Workflow	44
3.3	Hardware Part.....	45
3.3.1	Employed Microcontroller.....	45
3.3.2	Raspberry PI 4.....	46
3.3.3	Sensors.....	47
3.3.4	Actuators	51
3.4	Software Part.....	53
3.5	Conclusion.....	56
CHAPTER 4. GreenSense AI: Implementation and Results		57
4.1	Introduction.....	58
4.2	Data Acquisition	58
4.2.1	Sensor Measurement.....	59
4.2.2	ESP32 Microcontroller Data Collection and Transmission.....	62
4.2.3	Raspberry Pi Data Management	62
4.2.4	Node-RED Data Handling and Visualization.....	63
4.3	Climate Prediction.....	65
4.4	Insights and Recommendations.....	70
4.5	Results and Discussion.....	74
4.5.1	Prototype	74
4.5.2	Data Acquisition and Management	75
4.5.3	Climate Prediction	76
4.5.4	Visualization: Insights and Recommendations	76
4.6	Perspective and Future Directions	79
4.7	Conclusion.....	80
General Conclusion.....		81

Table of Contents

Appendices	83
A. Coding.....	83
B. Employed Hardware Components Specifications	86
Bibliography.....	89
Webography.....	91

List of Figures

Figure 1.1: IoT and Agriculture.....	9
Figure 1.2: Key IoT architectures.....	10
Figure 1.3: IoT general uses.....	13
Figure 1.4: Comparison of different IoT application protocols	18
Figure 1.5: MQTT publish-subscribe architecture	19
Figure 1.6: MQTT standard message format.....	20
Figure 2.1: ML learning paradigms.....	25
Figure 2.2: AI uses.....	26
Figure 2.3: Time Series components.....	32
Figure 2.4: RNN architecture	34
Figure 2.5: LSTM network architecture	36
Figure 2.6: An LSTM cell architecture.....	38
Figure 2.7: GRU network architecture.....	40
Figure 3.1: Description diagram of ESP32 Board.....	46
Figure 3.2: DHT22 sensor.....	48
Figure 3.3: SparkFun sensor	49
Figure 3.4: TSL2561 sensor	49
Figure 3.5: CCS811 sensor.....	50
Figure 3.6: Rain Drops level sensor	51
Figure 3.7: Employed lamp.....	52
Figure 3.8: Employed fan.....	52
Figure 4.1: Data Acquisition process	58
Figure 4.2: DHT22 sensor deployment.....	59
Figure 4.3: SparkFun sensor deployment	60
Figure 4.4: TSL2561 sensor deployment	60
Figure 4.5: CCS811 sensor deployment.....	61
Figure 4.6: Rain Drops sensor deployment	62
Figure 4.7: Interior node flow.....	63
Figure 4.8: Exterior node flow.....	64
Figure 4.9: Recommendations flow	64
Figure 4.10: GreenSense AI model architecture.....	67
Figure 4.11: Training and validation learning curves	68
Figure 4.12: Temperature predicted vs actual values.....	69
Figure 4.13: Learning rate progress during training.....	69
Figure 4.14: Actuators deployment	71
Figure 4.15: Interior node schema.....	71
Figure 4.16: Exterior node schema.....	72
Figure 4.17: Data flow in GreenSense AI	73

Figure 4.18: Interior sensing node	74
Figure 4.19: Inside view of the greenhouse.....	74
Figure 4.20: Exterior sensing node	75
Figure 4.21: Sensors readings	75
Figure 4.22: Interior node dataset	76
Figure 4.23: Temperature tab	76
Figure 4.24: Humidity tab	77
Figure 4.25: Soil moisture tab.....	77
Figure 4.26: RainDrops level tab	78
Figure 4.27: Light intensity tab	78
Figure 4.28: CO2 level tab	79
Figure 4.29: Perspective GreenSense AI architecture.....	80

List of Tables

Table 1.1: The employed IoT architecture.....	11
Table 1.2: Summary of Recent IoT in Greenhouses Studies	16
Table 2.1: Algorithms comparison.....	40
Table 3.1: Microcontrollers comparison.....	45
Table 4.1: GreenSense AI model architecture details.....	67

List of Abbreviations

UN DESA	United Nations Department of Economic and Social Affairs
FAO	Food and Agriculture Organization
IoT	Internet of Things
AI	Artificial Intelligence
CO₂	Carbon Dioxide
MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
AMQP	Advanced Message Queuing Protocol
HTTP	Hypertext Transfer Protocol
M2M	Machine-to-Machine
QoS	Quality of Service
ASI	Artificial Superintelligence
NLP	Natural Language Processing
RNN	Recurrent Neural Network
BPTT	Backpropagation Through Time
MLPs	Multi-Layer Perceptrons
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
VAR	Vector Autoregression
ARIMA	Autoregressive Integrated Moving Average
IDE	Integrated Development Environment
GUI	Graphical User Interface
RAM	Random Access Memory
I²C	Inter-Integrated Circuit
EEPROM	Electrically Erasable Programmable Read-Only Memory
CPU	Central Processing Unit

List of Abbreviations

SBC	Single-Board Computer
VIN	Voltage Input
GND	Ground
MCU	Microcontroller Unit
SQL	Structured Query Language
RDBMS	Relational Database Management System
ARM	Advanced RISC Machines
VNC	Virtual Network Computing
API	Application Programming Interface
JSON	JavaScript Object Notation
TCP/IP	Transmission Control Protocol/Internet Protocol
CSV	Comma-Separated Values
ReLU	Rectified Linear Unit
MSE	Mean Squared Error
MAE	Mean Absolute Error

General Introduction

According to the United Nations Department of Economic and Social Affairs (UN DESA), each second the world's population grows by nearly three more people, totaling 240,000 new individuals every day. According to UN DESA statistics, the global population is expected to reach 8 billion by 2025 and 9.8 billion by 2050 [1]. This implies an additional billion mouths to feed within the next decade and a substantial increase compared to the beginning of the 20th century.

With resources for sustainable food security already strained, the challenges are immense. Climate change is exacerbating these challenges by negatively impacting agricultural production on both a global and local scale. Farms are tasked with increasing food production while simultaneously preserving the environment; however, this goal cannot be achieved relying solely on traditional farming practices. In poorer countries, subsistence farming remains a labor-intensive and low-reward industry vulnerable to unpredictable environmental changes, economic fluctuations, and various other risk factors. Meanwhile, mechanized agriculture in developed nations has boosted output per unit of land; however, additional measures are necessary to meet future food demands sustainably.

Additionally, the utilization of freshwater resources in agriculture is significant, with no less than 70% of the planet's freshwater resources allocated primarily for irrigation purposes. However, as Food and Agriculture Organization (FAO) of the United Nations reports on global water resources and agricultural productivity shows, it's contradictory that only 20% of arable land is irrigated, yet these irrigated lands contribute to 40% of the world's food production, underscoring the crucial role of irrigation in global food security [2]. Nevertheless, these statistics serve as a stark reminder of the persistent insecurity within the agricultural sector.

Looking ahead to 2050, when the global population is projected to reach nine billion, the FAO predicts a 70% increase in food production will be necessary to meet growing demands [3]. This daunting task is compounded by agriculture's status as the largest consumer of freshwater resources, utilizing 70% of the world's freshwater. This underscores the urgent need to enhance agricultural sustainability and resource management practices to ensure future food security [4].

Fortunately, the advent of cutting-edge technologies, including Internet of Things (IoT) and Artificial Intelligence (AI), offers promising solutions. By connecting and integrating objects, individuals, information, and systems, IoT facilitates intelligent production and services. Coupled with AI, these technologies enable advanced data analytics and predictive modeling, propelling farming into a new era of precision agriculture and enhanced productivity.

In this project, the aim is to develop a smart system applicable to any greenhouse. This system harnesses the power of both IoT and AI. The IoT aspect encompasses the hardware and software functionalities related to sensor connectivity, data acquisition and data transmission within the

system. The AI aspect then analyzes this data using various methods, enabling precise decision-making for current conditions and predictive insights for the near future, offering actionable recommendations for farmers. These operations occur within milliseconds and rely on several factors. The ultimate goal is to advance agriculture, particularly greenhouse cultivation, to help farmers achieve better crop growth monitoring and selection systems.

This project is organized in four main chapters:

Chapter One introduces the IoT component of the project, providing an overview of IoT systems and their implementations in agriculture. It discusses the benefits and drawbacks, presents the general architecture of IoT systems, and covers in detail the most important communication protocols used in the system.

Chapter Two shifts focus to another component of the project, AI. It defines AI and provides its theoretical foundations to offer a comprehensive understanding of the field, highlighting its profound impact across various sectors, particularly agriculture. Additionally, the chapter delves into time series analysis, outlining its theoretical basis and emphasizing the forecasting models commonly used, thus paving the way for the chosen model architecture.

Chapter Three provides an overview of the project's system, detailing the hardware and software components. It discusses the rationale behind selecting specific hardware devices and introduces essential software tools.

Chapter Four transitions to the implementation phase of the project, providing an overview of the data flow within the system up to the user. The chapter details the processes involved in system construction, including installation procedures, service descriptions, and other functionalities. Finally, it presents the achieved results of the proposed solution along with a comprehensive explanation, showcasing its potential impact.

CHAPTER 1.

Internet of Things and Agriculture

1.1. Introduction

Smart agriculture has undergone significant evolution in recent years, driven by the integration of IoT technologies that offer substantial advancements in farming practices. This chapter delves into this evolution, beginning with an understanding of the IoT, covering its definition, concepts, and architecture overview. It then explores the growing importance of IoT in agriculture and its pivotal role in shaping the industry's future. The discussion extends to an overview of IoT applications across sectors, emphasizing its transformative potential while discussing their advantages and disadvantages. Lastly, the chapter provides a detailed overview and comparative analysis of IoT protocols currently employed in the field.

1.2. Understanding IoT

IoT refers to a set of technologies and infrastructures that enable various objects to connect to the internet and operate in an interconnected manner. These connected objects can be remotely controlled using a computer, smartphone, or tablet.

The concept of the IoT encompasses all objects and physical devices that possess a digital identity. These objects can range from common household appliances such as televisions, refrigerators, washing machines, and heating systems, to more complex devices and systems such as vehicles (planes, autonomous cars), and urban lighting.

In agriculture and greenhouses, IoT plays a crucial role in enhancing productivity and efficiency by enabling monitoring and control of environmental conditions. This technology allows farmers to optimize resource use, improve crop yield, and implement sustainable practices more effectively.



Figure 1.1: IoT and Agriculture [5]

1.2.1. IoT Architecture Overview

The IoT architecture comprises several connected IoT system components to ensure that object data generated by sensors is collected, stored, and processed in big data warehouses, and that object actuators execute commands sent via a user application. It is a framework that defines the physical components, functional organization, network configuration, operational procedures, and data formats to be used. Although there is no universally accepted single IoT architecture, the most basic and widely accepted format is a four-layer IoT architecture.

“Figure 1.2” outlines the primary IoT architectures.

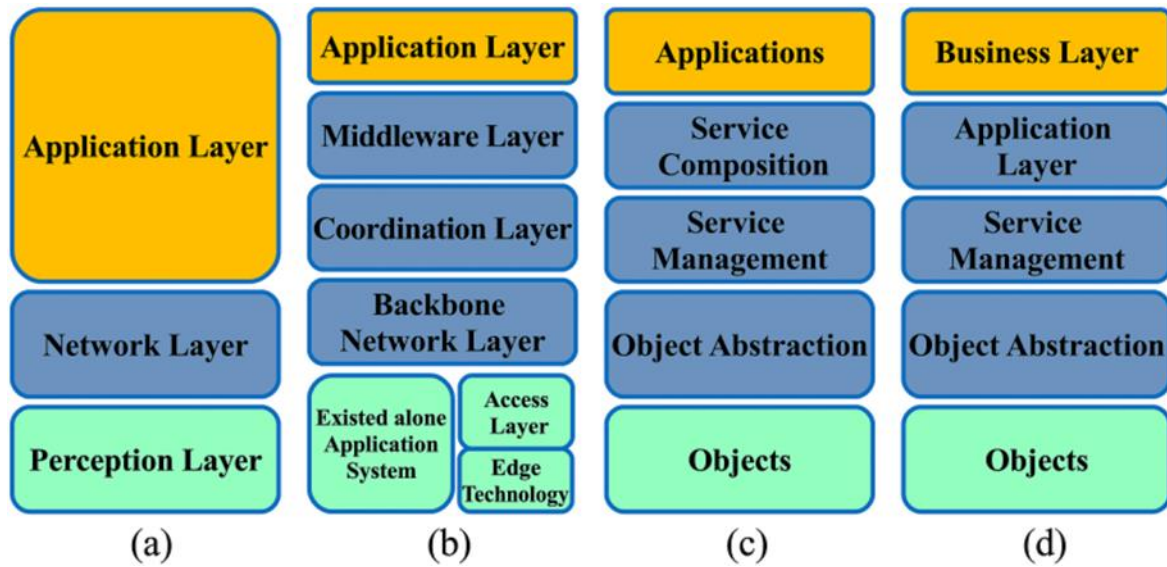


Figure 1.2: Key IoT architectures [6]

For the project, the adopted architecture is the (a) model which is based on three layers:

Perception Layer: This layer is responsible for converting analog signals into digital data and vice versa. The initial step of any IoT system encompasses a diverse array of objects that act as a bridge between the physical and digital worlds. These objects vary in shape and size, ranging from tiny silicon chips to large vehicles, each tailored to their specific functions.

Network Layer: The data collected by all these devices needs to be transmitted and processed. This is the role of the network layer, which connects these devices to other smart objects, servers, and network devices. It also manages the transmission of all data. Communications between devices and cloud services or gateways involve various technologies, including Ethernet, Cellular Networks, Wi-Fi.

Application Layer: The application layer is what the user interacts with. It is responsible for providing specific application services to the user. Currently, applications can be built directly on IoT platforms that offer software development infrastructure with ready-to-use tools for data exploration, advanced analysis, and data visualization

“Table 1.1” simplifies the IoT architecture adopted.

Application layer	IoT Applications
	Application Support
Network layer	Local and Wide Area Network
	Core Network
	Access Network
Perception layer	Perception Network
	Perception Node

Table 1.1: The employed IoT architecture

1.3. IoT Today: The Tipping Point

Assuming that IoT devices exclude phones, tablets, laptops, and desktop computers, their number has dramatically increased between 2010 and 2020, growing by approximately 1000%. Nearly 10 billion IoT devices have been deployed and interconnected over the past decade.

To put this evolution into perspective, in 2010, IoT devices accounted for 10% of the total connected devices. By 2020, this share had exceeded 50%, marking a tipping point. IoT devices are now officially more numerous than non-IoT devices.

According to a global market analysis report, IoT generated a global revenue of over \$300 billion in 2019, with rapid growth. Many IoT devices incorporate low-power microcontrollers. Over 28 billion of these microcontrollers were shipped in 2018, and it is estimated that by 2020, over 250 billion were in use worldwide [7]. Although not all microcontrollers are network-connected, an increasing number of deployments combine microcontrollers running increasingly complex code with a direct or indirect connection to a network. Most connect via wireless networks such as personal, local, wide area, or cellular networks.

These deployments are rapidly expanding across various sectors, notably including automotive applications, industrial automation, healthcare, and smart home devices.

1.4. IoT Applications in General

Intel and its partner ecosystem assist businesses in leveraging IoT to address long-standing challenges specific to each industry. Its portfolio of open and scalable solutions enables the rapid development of connected devices, data collection, and insights generation to reduce costs, enhance productivity, and increase revenue. There are several application domains that will be impacted by the emerging Internet of Things. Applications can be classified based on network availability, coverage, scale, heterogeneity, repeatability, user participation, and impact.

Connected Accessories: Connected accessories represent one of the most visible aspects of IoT for everyday users. These include activity monitors, smartwatches, smart glasses, virtual reality headsets, etc.

Smart Homes: A smart home system connects to devices to automate specific tasks and is typically remotely controlled. Examples of IoT devices in smart homes include wireless kitchen appliances, mood-detecting music systems, smart lighting, motorized blinds, automated doors and windows, smart electric meters, etc.

Smart Cities: Smart cities utilize IoT devices such as sensors and connected meters to collect and analyze data. This data can be used to enhance infrastructure, public services, and other facilities.

Autonomous Vehicles: Autonomous vehicles are typically equipped with IoT-based technology systems that share information about the vehicle itself and the route it takes. Data regarding traffic, navigation, the external environment, and more are collected and analyzed by the vehicle's computer systems to enable autonomous driving.

Telehealth: Telehealth, sometimes referred to as telemedicine, involves delivering healthcare services through technology. IoT is integral to telehealth, and the acronym IoMT is sometimes used to specifically refer to the Internet of Medical Things. For example, IoMT facilitates remote medical diagnosis, digital communication of medical imaging, and video consultations with specialists.

Smart Agriculture: Smart agriculture involves using digital technology to enhance agricultural efforts. For instance, farmers can utilize sensors, cameras, and other connected devices to improve their farm's overview and adapt operations to enhance yield.

This list is indicative rather than exhaustive. Across various sectors, IoT is revolutionizing operations and methodologies. IoT devices range widely, including smartphones, smart refrigerators, smartwatches, activity monitors, smart fire alarms, smart door locks, smart bicycles, medical sensors, smart security systems, and virtual assistants such as Alexa and Google Home, among others.

“Figure 1.3” provides a summary of the key applications of IoT.

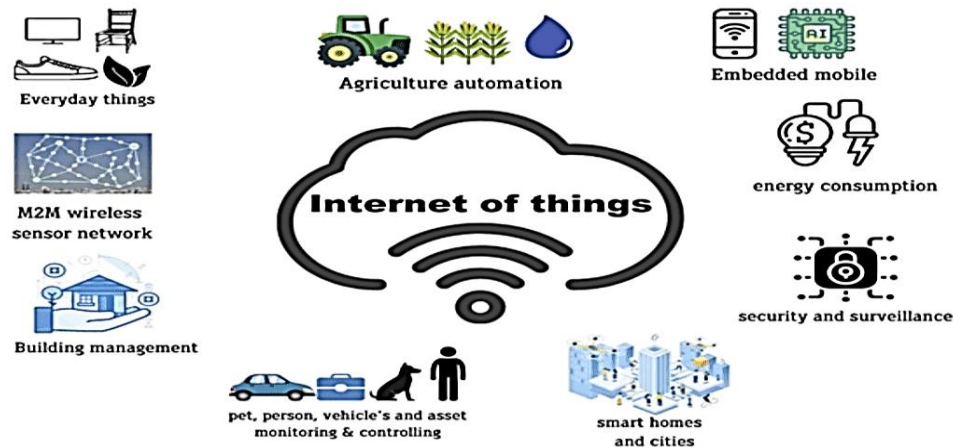


Figure 1.3: IoT general uses [8]

1.5. IoT in Agriculture

The IoT is revolutionizing agricultural practices by enabling precise monitoring and control of critical climatic factors through devices such as sensors and automation systems. This technological advancement facilitates the collection and analysis of diverse environmental parameters, significantly enhancing productivity and efficiency in farming. By optimizing conditions for crop growth and resource management, IoT plays a pivotal role in modernizing agricultural operations and ensuring sustainable practices. Here are the key points:

Precision Farming: IoT devices gather data on soil moisture, temperature, and nutrient levels, among other parameters, allowing for precise management of crops. Drones and weather stations provide real-time insights to optimize planting, watering, and harvesting schedules.

Smart Irrigation Systems: IoT-enabled irrigation systems use soil moisture sensors and weather data to automate and optimize watering schedules, reducing water waste and ensuring crops receive the adequate amount of water.

Crop Monitoring: Sensors and drones monitor crop conditions, detecting issues such as nutrient deficiencies, pest infestations, and diseases early, allowing for timely interventions.

Supply Chain Optimization: IoT devices enhance traceability and reduce waste by monitoring storage and transportation conditions. This ensures the freshness of agricultural produce and efficient inventory management.

From the above points, IoT's integration into agriculture not only boosts productivity through precise monitoring and automation but also promotes sustainable practices by optimizing resource use and enhancing crop health, marking a significant advancement in modern farming techniques.

1.6. IoT in Greenhouses

Greenhouses, as enclosed agricultural environments, benefit similarly from IoT technologies. Here's how IoT is implemented in greenhouses:

Environmental Monitoring: IoT sensors are strategically deployed throughout the greenhouse to monitor essential environmental factors such as temperature, humidity, light intensity, and Carbon Dioxide (CO₂) levels. These sensors continuously collect data, which is then analyzed in real-time. This data helps in maintaining optimal growing conditions for various crops.

Automated Climate Control: Based on the data collected by IoT sensors, automated systems can adjust environmental conditions within the greenhouse. For example, automated ventilation systems regulate airflow and temperature. This automation ensures that crops receive consistent and optimal growing conditions.

Water and Nutrient Management: IoT-enabled systems monitor soil moisture levels and nutrient concentrations in real-time. This data enables precise control over irrigation and fertilization, ensuring that plants receive water and nutrients precisely when and where they are needed most.

Pest and Disease Monitoring: IoT devices such as sensors and cameras can detect early signs of pests and diseases within the greenhouse. This early detection allows prompt actions such as targeted treatments or adjustments to environmental conditions, minimizing crop damage and reducing reliance on chemical interventions.

Energy Efficiency: IoT technologies can optimize energy consumption within the greenhouse by intelligently controlling heating, cooling, and lighting systems based on real-time data and predictive analytics. This efficiency helps reduce operational costs and environmental impact.

Overall, IoT in greenhouses represents a significant advancement in agricultural technology, equips growers with the tools they need to enhance productivity, improve crop quality, and adopt more sustainable practices in controlled environment agriculture.

1.6.1. Related Work

Recent advancements in IoT applications for agriculture and greenhouse management have sparked significant improvements in crop yields, resource optimization, and overall crop quality. “Table 1.2” highlights a selection of key projects and initiatives, showcasing the diverse applications and transformative potential of IoT in modernizing farming practices.

Related paper	Application Domain	Sensors	Actuators	Communication technologies	Conclusions
[9]	- Monitoring - Controlling - Detecting	- Temperature - Soil Moisture - Humidity	- Irrigation - Fertilization	- Wi-Fi - Zigbee	- Excellent reliability - Integration difficulties - Low power consumption - Low-range connectivity - Limitations in terms of memory and power backup requirements.
[10]	- Monitoring	- Temperature - Soil Moisture - Sunlight - Humidity sensor	- Humidifier - Heater - Cooling fan	- WSN	- Fast response time - Complexity in the system - Resource constraints.
[11]	- Monitoring - Predicting	- Temperature - Humidity	None	- LPWAN	- High cost - Long range area - Great flexibility - Limited Data Throughput - Higher Latency - Low power consumption
[12]	- Monitoring - Controlling	- Temperature - Humidity	- Water irrigation	- WSN - Zigbee - GSM/GPRS	- Excellent reliability - Support both long range area and low range area
[13]	- Monitoring	- Soil moisture - CO2	None	- Wi-Fi - 4G/LTE	- Excellent reliability

		- Light intensity - Temperature - Humidity - Pressure sensor			- Support both long range area and low range area - Great flexibility - Low power consumption - Fast response time
[14]	- Monitoring	- Temperature - Humidity - Soil and air quality	None	- MQTT	- Stand out where network bandwidth and power consumption are critical factors - Require more resources

Table 1.2: Summary of Recent IoT in Greenhouses Studies

The project's technology choices are guided by its specific requirements. Given that the microcontroller and Raspberry Pi will always be co-located within the greenhouse, and considering that bandwidth and latency are not significant concerns in a local network, Wi-Fi over Transmission Control Protocol/Internet Protocol (TCP/IP) was chosen as the primary communication protocol between them. This ensures reliable and efficient data transfer within the greenhouse.

For data transfer between the Raspberry Pi inside the greenhouse and the server hosting the web server, typically a central Raspberry Pi, MQTT was chosen due to its advantageous features. It is particularly well-suited for scenarios where network bandwidth and power consumption are critical factors. Communication between the Raspberry Pi hosting the web server and the user is also facilitated through Wi-Fi. Actuators are controlled via MQTT.

Furthermore, the six selected sensors, which are temperature, humidity, soil moisture, raindrops, CO₂, and light intensity, are commonly used in similar monitoring projects to meet the demands of AI analysis. These sensors were chosen based on their proven effectiveness in similar applications, ensuring that the monitoring system is robust and reliable.

1.7. Advantages and Disadvantages of IoT in Agriculture

IOT offers significant benefits for agriculture and greenhouse management, yet it also entails challenges and limitations,

1.7.1. Advantages

Automation: IoT automates tasks like irrigation, lighting, and climate control in greenhouses, optimizing daily operations and enhancing efficiency. Automated systems adjust conditions in real-time, ensuring optimal growth environments for crops.

Monitoring: IoT enables remote monitoring of various environmental factors. Real-time data collection and analysis are crucial for optimal growing conditions.

Accuracy: IoT-based systems enhance process accuracy. For example, soil sensors provide precise moisture data, ensuring crops receive the optimal amount of water, leading to higher yields and better-quality produce.

Innovation: IoT drives innovation in agriculture. For instance, smart sensors and drones improve crop management by providing detailed insights into plant health, pest activity, and soil conditions. Automated systems can also adjust the greenhouse environment based on real-time data, leading to better crop management and increased productivity.

1.7.2. Disadvantages

Security: The interconnected nature of IoT devices poses cybersecurity risks. Each connected device can be a potential vulnerability. In agriculture and greenhouses, a cyberattack could disrupt operations, leading to crop damage or loss. Ensuring robust security measures for IoT systems is essential to protect sensitive data and maintain operational integrity.

Privacy: IoT devices in agriculture often collect data on farming practices and crop conditions. This data, if misused, could lead to privacy concerns for farmers. Protecting the privacy of farming data requires stringent data protection policies and compliance with regulations.

Complexity: Implementing and managing IoT systems in agriculture is complex and requires specialized skills. Setting up IoT devices involves integrating various technologies, ensuring compatibility, and maintaining the network. Ongoing maintenance and troubleshooting demand expertise, which can be a barrier for some farmers.

Costs: The initial investment in IoT infrastructure for agriculture can be high. Costs include purchasing IoT devices, setting up the network infrastructure, and investing in data storage and processing capabilities. While IoT can lead to long-term savings and increased productivity, the upfront expenses can be a significant barrier to adoption for small-scale farmers.

This balanced perspective underscores both the potential benefits and challenges associated with adopting IoT technologies in agriculture and greenhouses. By understanding these aspects, stakeholders can make informed decisions about integrating IoT into their operations.

1.8. IoT Communication Protocols

IoT communication protocols form the backbone of communication standards that enable seamless interaction among devices and systems. These protocols define how data is transmitted, ensuring reliability, efficiency, and scalability in diverse IoT applications [15].

Various communication protocols are utilized in IoT projects, including Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), these three are the most popular. CoAP, is designed for simple, constrained environments and operates on a request-response model similar to Hypertext Transfer Protocol (HTTP). However, it is less suitable for scenarios requiring continuous data streams like Greenhouse monitoring. AMQP, while robust and capable of handling complex routing scenarios and message queuing within enterprise environments, tends to introduce higher network overhead and complexity compared to MQTT, making it less suitable for resource-constrained IoT devices and applications requiring lightweight communication and scalability.

In contrast, MQTT provides more efficient data transmission and better scalability for large networks of devices. It also has lower network overhead and is easier to implement, making it ideal for IoT devices with limited resources. Therefore, for smart greenhouse projects, MQTT offers a balanced approach by providing lightweight, scalable communication and reliable data transmission, making it a preferred protocol for such IoT applications.

“Figure 1.4” present a comprehensive comparison of protocols available for IoT usage

Application Protocol	RESTful	Transport	Publish/Subscribe	Request/Response	Security	QoS	Header Size (Byte)
COAP	✓	UDP	✓	✓	DTLS	✓	4
MQTT	✗	TCP	✓	✗	SSL	✓	2
MQTT-SN	✗	TCP	✓	✗	SSL	✓	2
XMPP	✗	TCP	✓	✓	SSL	✗	-
AMQP	✗	TCP	✓	✗	SSL	✓	8
DDS	✗	TCP UDP	✓	✗	SSL DTLS	✓	-
HTTP	✓	TCP	✗	✓	SSL	✗	-

Figure 1.4: Comparison of different IoT application protocols [16]

1.8.1. MQTT Definition

MQTT or Message Queuing Telemetry Transport is real-time data transmission protocol. It is built on TCP protocol, it was developed by IBM in 1999, it is a very lightweight communication protocol to be used especially in Machine-to-Machine (M2M) and places where the connection is weak [17]. MQTT is not just a protocol; it's a conduit to innovation. Whether you're shaping the future of smart cities, revolutionizing healthcare through connected devices, or crafting the next generation of industrial automation solutions, MQTT will be your steadfast ally.

1.8.2. MQTT Architecture

MQTT, relies on several fundamental components that define its efficient and reliable communication protocol in IoT and beyond. These components ensure streamlined messaging, efficient data transfer, and robust connectivity management, making MQTT a cornerstone in modern IoT implementations. These components are:

MQTT Broker: It is a crucial component of the MQTT protocol architecture, enabling communication between IoT devices by receiving published messages from clients and ensuring they are accessible to other subscribing clients. With its robust capabilities, an MQTT broker can manage numerous simultaneous connections and process millions of messages efficiently.

MQTT Client: An MQTT client is a device or application that connects to an MQTT broker to either send messages (act as a publisher) or receive messages (act as a subscriber) on specific topics within an MQTT-based communication network [17].

“Figure 1.4” illustrates an example of MQTT communication.

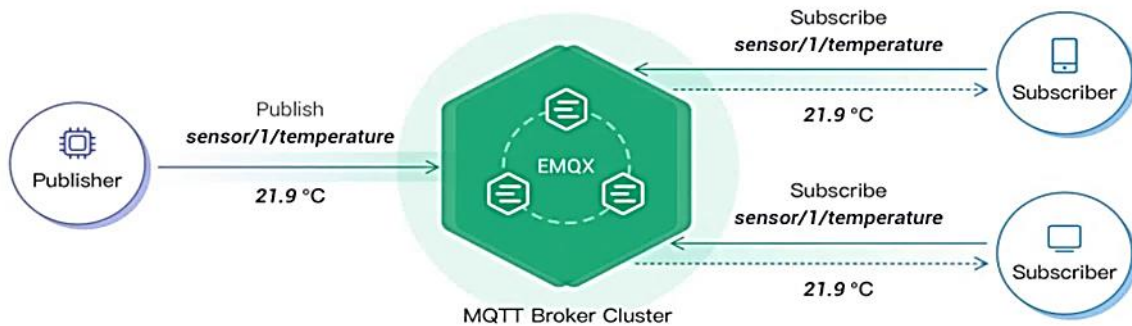


Figure 1.5: MQTT publish-subscribe architecture [18]

1.8.3. MQTT Message Format

The MQTT message format comprises four essential components: the control header, packet length, variable length header, and payload. The control header contains critical metadata such as the message type and Quality of Service (QoS), offering control over message delivery and acknowledgment. The packet length field indicates the overall length of the MQTT message,

including all its components. Following this, the variable length header holds specific message-related details like topic names or message identifiers, varying in size based on the message type. Finally, the payload carries the actual data being transmitted, which could range from sensor readings to commands or any application-specific information. Together, these components form a structured and efficient messaging system ideal for IoT applications, ensuring reliable communication across constrained networks while minimizing bandwidth usage and latency.

“Figure 1.5” displays a packet structure of MQTT.

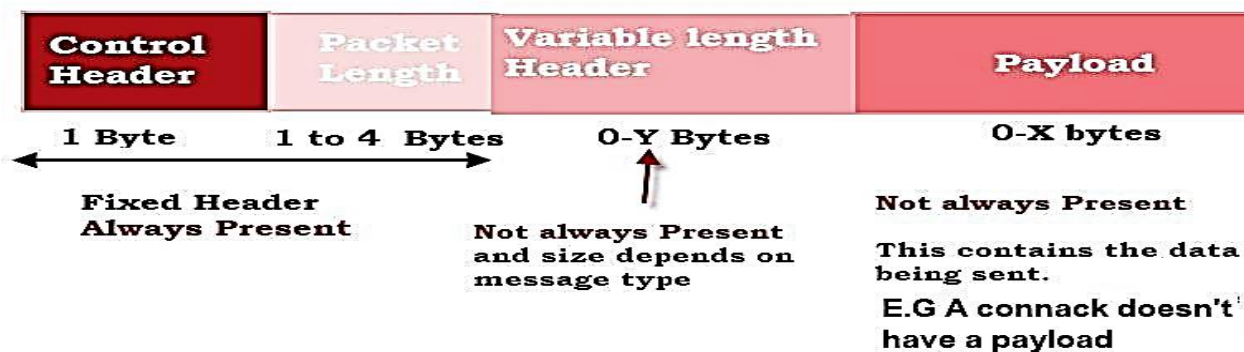


Figure 1.6: MQTT standard message format [15]

1.9. Conclusion

In conclusion, the IoT offers a promising technology for enhancing connectivity and data exchange among physical objects, particularly within greenhouses. This chapter has examined its definition and architecture, with a focus on MQTT as a common IoT communication protocol, and explored its applications both within and beyond agriculture, including its advantages and disadvantages.

As we move forward, the next chapter will explore the integration of AI into smart greenhouse systems, aiming to further innovate and enhance productivity in agricultural technology. AI, combined with IoT, promises to revolutionize the industry by providing advanced analytics, predictive capabilities, and autonomous decision-making, further advancing modern farming techniques.

CHAPTER 2.

Artificial Intelligence and Agriculture

2.1. Introduction

Sustainable development in agriculture is essential to meet the demands of a rapidly growing global population and the increasing need for food. Agriculture must not only enhance food production but also prioritize environmental preservation. Sustainable farming practices significantly contribute to this goal by promoting efficient resource use, reducing waste, and minimizing environmental impact [19],[20]. Integrating advanced technologies, particularly AI, can further accelerate and enhance these sustainable efforts.

AI holds the potential to revolutionize agriculture by optimizing processes and enhancing decision-making. In the context of greenhouse management, AI can provide accurate climate predictions, improve resource allocation, and elevate both crop yield and quality [21]. By implementing AI-driven solutions, farmers can adopt more sustainable farming practices, thus making substantial contributions to global food security.

This chapter discusses the transformative role of AI in agriculture, specifically focusing on predicting climate parameters within greenhouses. It explores AI fundamentals, including its definition and theoretical concepts, before delving into its diverse applications in agriculture, encompassing both broad and specialized domains such as greenhouse management.

Additionally, the chapter delves into another crucial facet in agriculture: time series analysis and forecasting for predictive purposes. It emphasizes the use of advanced AI methodologies, particularly Recurrent Neural Networks (RNNs), within this domain. Following the discussion on time series, the chapter further elucidates the rationale behind selecting AI models for greenhouse climate prediction within our proposed solution.

2.2. Artificial Intelligence

AI is the science and engineering of developing intelligent machines, especially computer systems, that simulate human intelligence processes. These systems are designed to think, learn, and make decisions similarly to humans, using algorithms and extensive datasets to identify patterns, predict outcomes, and perform tasks that typically require human cognition [22]. Such tasks include visual perception, speech recognition, decision-making, language translation, reasoning, learning, perceiving, and problem-solving. While AI often aims to understand and replicate human intelligence, it is not restricted to biologically observable methods.

2.2.1. Theoretical Foundations of AI

This section delves into the foundational theories and historical evolution that have shaped AI into a pervasive force in modern technology.

Brief history of AI

The origins of AI can be traced back to the mid-20th century with visionaries like Alan Turing, who introduced the concept of machines capable of simulating human intelligence. The field was officially founded in 1956 at the Dartmouth Summer Research Project on Artificial Intelligence. Early developments in the 1960s and 70s included AI systems like DENDRAL and the robot Shakey, but the mid-70s brought an "AI winter" due to critiques and funding issues. The 1980s experienced a revival with expert systems and projects like the Japanese Fifth Generation Computer, but another AI winter followed in the late 80s. The 1990s and 2000s marked a resurgence with machine learning techniques, such as neural networks, leading to successes like IBM's Deep Blue defeating chess champion Garry Kasparov. The 2010s brought explosive growth in AI, driven by advancements in computing power, big data, and deep learning, resulting in AI systems surpassing human performance in various tasks and being widely adopted across various industries [22].

Types of AI

AI can be categorized into various types based on capabilities. Here are the main types:

Narrow AI (Weak AI): Designed to perform specific tasks or solve particular problems. These systems are highly specialized and cannot perform tasks outside their specific domain. Examples include virtual personal assistants like Siri and Alexa, recommendation systems, and facial recognition software.

General AI (Strong AI): Hypothetical systems with human-level intelligence across a wide range of tasks and domains. Such AI would have generalized learning and reasoning abilities, enabling them to understand, learn, and apply knowledge in ways that mimic human cognitive abilities. General AI remains theoretical and is yet to be achieved.

Artificial Superintelligence (ASI): Theoretical level of AI that surpasses human intelligence in all aspects, from creativity to problem-solving. ASI would possess capabilities far beyond what humans can comprehend and may potentially lead to unpredictable outcomes. This level of AI remains speculative and is the subject of much debate and ethical consideration [23].

Recognizing these diverse AI capabilities offers insight into their current applications and future potential, guiding us in utilizing AI's benefits while responsibly managing its challenges.

Core Concepts

In the realm of AI, several core concepts lay the groundwork for understanding and development. Let's explore these concepts, arranged logically to facilitate understanding:

Data and Big Data: Data serves as the fuel that powers AI systems, where its quality, quantity, and diversity significantly impact performance. Big data refers to large volumes of structured and unstructured data that AI models utilize for training and inference.

Algorithms: Serving as the backbone of AI systems, algorithms provide step-by-step procedures or formulas for solving diverse problems, they are used to process data, extract insights, learn from it, and make predictions or decisions. Algorithms can range from simple rule-based approaches to complex mathematical models.

Neural Networks: Inspired by the structure and function of the human brain, neural networks are a fundamental component of deep learning, a subset of AI. Comprising interconnected nodes called neurons organized into layers, these networks process information hierarchically. Each neuron applies a mathematical operation to its input and passes the result to the next layer, mimicking cognitive functions to achieve complex tasks.

Training and Learning: AI systems learn from data through training processes, adjusting algorithm parameters to minimize prediction errors. Learning occurs via supervised, unsupervised, or reinforcement-based methods, enhancing model adaptability and refinement.

Model Evaluation and Validation: Post-training, AI models need to be evaluated and validated to ensure their accuracy, reliability, and generalizability. Evaluation involves assessing the model's performance using various metrics, while validation methods help assess how well the model performs on unseen data.

Understanding these fundamental concepts provides a robust framework for navigating the complexities of AI, empowering individuals to unleash its potential in real-world applications.

Learning Paradigms

Learning paradigms refer to the various approaches and methodologies through which AI systems acquire knowledge, learn patterns, and make decisions. These paradigms encompass different techniques and strategies for teaching machines to understand and adapt to their environments. Here are some key learning paradigms in AI:

Supervised Learning: This paradigm involves training AI models on labeled data, where inputs are paired with corresponding output labels. The model learns to map inputs to outputs by minimizing the difference between its predictions and the true labels. This approach is widely used in tasks such as classification and regression.

Unsupervised Learning: This paradigm involves training AI models on unlabeled data, aiming to uncover underlying patterns or structures within the data. Without predefined output labels, the model must identify meaningful representations or clusters in the input data. It is commonly used in tasks such as clustering, dimensionality reduction, and anomaly detection.

Reinforcement Learning: This paradigm involves training AI models to make a sequence of decisions by rewarding good decisions and penalizing bad ones. the model explores different actions and learns from the rewards obtained. This approach is prevalent in applications such as game playing, robotics, and autonomous systems [24].

“Figure 2.1” presents the three main learning paradigms along with some of their applications

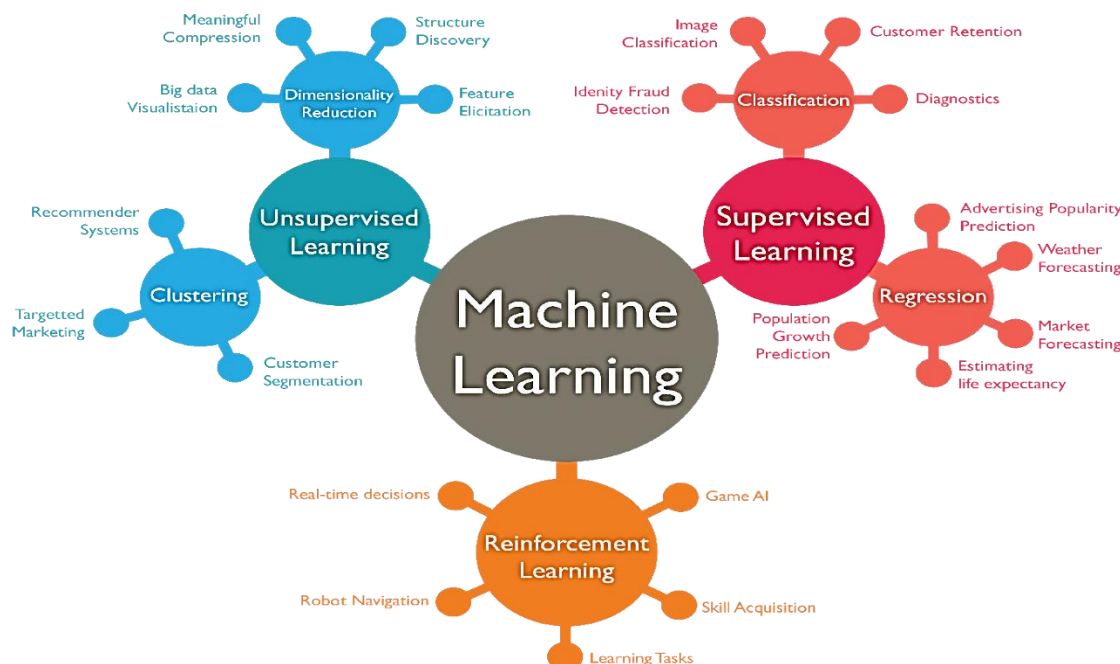


Figure 2.1: ML learning paradigms [25]

These are the primary paradigms in AI, each with distinct methodologies and applications. Additionally, other paradigms like semi-supervised learning, self-supervised learning, and transfer learning are also important and contribute significantly to the advancement of AI technology.

2.3. Key Areas and Applications of AI

AI encompasses a range of specialized fields, each contributing to different applications and industries. Here are some key areas and their applications:

Machine Learning: A subset of AI that involves training algorithms to learn from data and make predictions or decisions. It is widely used in predictive analytics for forecasting events, recommendation systems for suggesting products or content, and anomaly detection for identifying unusual patterns.

Deep Learning: A specialized branch of machine learning, deep learning uses neural networks with multiple layers to model complex data patterns. This technology powers applications such as image and speech recognition, and autonomous vehicles for real-time decision-making.

Natural Language Processing (NLP): A field of AI focused on the interaction between computers and human language. Key applications include chatbots for automated customer service, sentiment analysis for understanding text emotions, and language translation for converting text or speech between languages.

Computer Vision: This field enables machines to interpret and make decisions based on visual data. Applications include facial recognition for identifying individuals, object detection for locating items in images or videos, and medical image analysis for diagnosing diseases from scans.

Robotics: AI integration in robotics involves designing and using robots to perform various tasks. In manufacturing, robots automate repetitive tasks; in healthcare, they assist in surgeries and patient care; and in service industries, they perform tasks like cleaning and delivery [26].

"Figure 2.2" summarizes these and other key fields and industries where AI is applied.

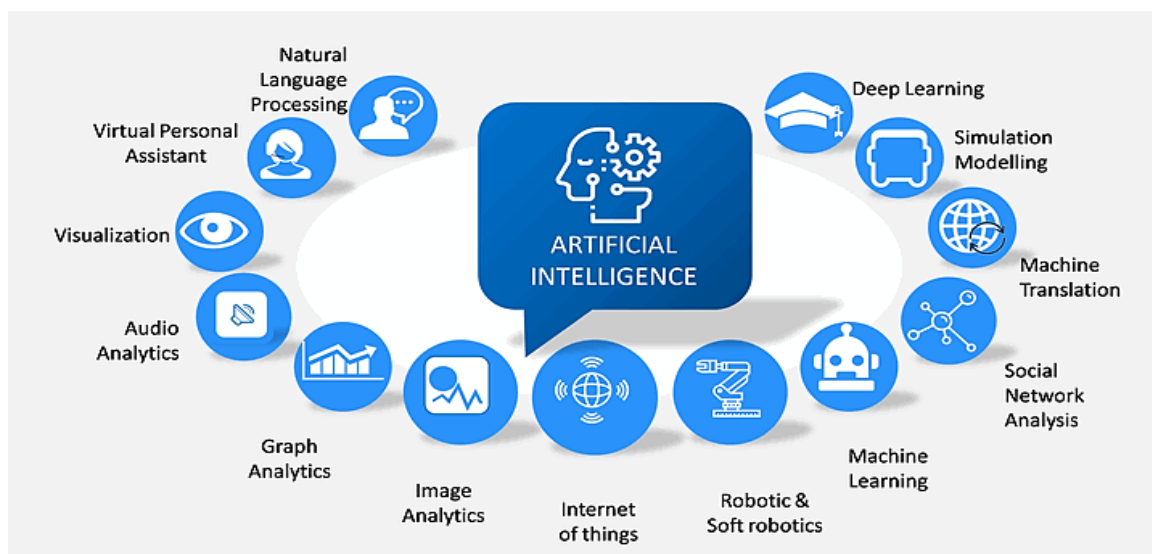


Figure 2.2: AI uses

These key areas highlight the diverse applications and transformative potential of AI across multiple sectors, underscoring its role in shaping the future of technology and industry.

2.4. Importance of AI in Modern Technology and Industry

AI has become a cornerstone of modern technology and industry due to its ability to enhance efficiency, accuracy, and decision-making processes. Across various sectors, AI-driven technologies are revolutionizing traditional practices:

Healthcare: AI is transforming healthcare by diagnosing diseases with high accuracy, personalizing treatment plans based on individual patient data, and efficiently managing extensive patient records. Applications like predictive analytics help in early disease detection, while AI-driven tools assist in surgical procedures and drug discovery.

Finance: In the financial sector, AI is pivotal in detecting fraud through the analysis of transaction patterns, conducting algorithmic trading with rapid, data-driven decisions, and offering personalized financial advice to customers based on their spending habits and financial goals. Additionally, AI aids in credit scoring and risk management.

Transportation: Autonomous vehicles, one of the most prominent advancements in AI, rely on machine learning algorithms to navigate and make real-time decisions. AI is also crucial in traffic management systems, optimizing traffic flow, reducing congestion, and enhancing road safety.

Manufacturing: AI improves manufacturing processes through predictive maintenance, which anticipates equipment failures and schedules timely repairs, thereby reducing downtime. It also enhances quality control by identifying product defects and optimizes supply chains through efficient resource management and demand forecasting.

Customer Service: AI-powered chatbots and virtual assistants are revolutionizing customer service by delivering instant responses to inquiries, resolving common issues, and boosting overall service efficiency. These technologies allow businesses to provide 24/7 support and personalized customer interactions, significantly enhancing user experience and customer satisfaction [27].

AI's ability to swiftly and accurately process and analyze large amounts of data is driving innovation across various industries, making it a cornerstone of modern technological advancements and industrial efficiency.

2.5 AI and Agriculture

AI's profound impact across various sectors makes it clear that it has the potential to revolutionize agriculture as well. By leveraging AI's capabilities, the agricultural sector can tackle pressing challenges such as optimizing resource use, improving crop yields, and enhancing sustainability, especially in the face of challenges like climate change and population growth [28]. This rapidly evolving field promises to transform traditional farming practices and significantly enhance food production. In the following section, we will explore the nature of data used in AI applications in agriculture and examine how these technologies are being practically applied to shape the future of farming.

2.5.1 Nature of Data Utilized in Agricultural AI:

In the realm of AI, data quality stands as a fundamental pillar determining the efficacy of models deployed. The precision and nature of data utilized significantly influence the performance of AI algorithms. In the agricultural sector, this understanding of nature of data holds particular importance. This section explores the various types of data used in AI within agriculture, particularly within the context of agricultural greenhouses:

Weather Data: Vital for understanding environmental conditions impacting crop growth, weather data includes parameters such as temperature, humidity, precipitation, and wind speed, sourced from weather stations, satellites, and IoT sensors. When considering greenhouses specifically, these metrics can be measured both indoors and outdoors.

Soil Data: Offering comprehensive insights into soil properties, which are crucial for optimizing soil management practices, soil data encompasses factors such as composition, moisture content, nutrient levels, and pH levels. This data is often gathered through soil sampling and analysis techniques.

Crop Growth Data: Essential for monitoring and predicting crop development, crop growth data encompasses various metrics such as plant height, leaf area index, flowering stages, and yield projections. This data enables greenhouse operators to make informed decisions regarding crop management strategies and resource allocation.

Pest and Disease Data: Early detection and mitigation of pests and diseases are critical for preserving crop health and minimizing yield losses. This data involves monitoring occurrences and outbreaks, obtained through visual inspection, sensor networks, and image analysis, enabling greenhouse managers to implement timely interventions and pest control measures.

Market Data: Influential in strategic decision-making and market analysis, market data encompasses trends, pricing information, demand forecasts, and consumer preferences. By leveraging market data, greenhouse operators can tailor their production and marketing strategies to meet consumer demand and maximize profitability [28].

2.5.2 Application of AI Technologies in Agriculture

The integration of AI into agricultural practices marks a transformative shift in farming methodologies. This section explores the broad applications of AI across agriculture, followed by a focused examination of its utilization within greenhouse operations.

Broad Application of AI Technologies in Agriculture:

AI technologies are revolutionizing agriculture by offering a broad range of applications that enhance efficiency, sustainability, and productivity across various farming practices. These technologies leverage data analytics, machine learning, computer vision, and automation to optimize decision-making processes and improve outcomes in the agricultural sector. Here are some key aspects of the broad application of AI technologies in agriculture:

Precision Agriculture: AI enables precision agriculture by providing farmers with real-time insights into crop health, soil conditions, and weather patterns. This data-driven approach allows for targeted interventions such as optimized irrigation, fertilization, and pest management, leading to increased yields and reduced resource wastage.

Crop Monitoring and Management: AI tools like drones, sensors, and satellite imagery are used to monitor crops, detect diseases, pests, and weeds, and assess overall crop health. By analyzing this data, farmers can make informed decisions regarding planting, harvesting, and crop protection strategies.

Smart Farming Equipment: AI-integrated machinery such as autonomous tractors, drones, and robots are transforming farming operations. These technologies can perform tasks like planting, spraying, and harvesting with precision and efficiency, reducing labor requirements and operational costs.

Data-Driven Decision Making: AI algorithms process vast amounts of agricultural data to provide actionable insights for farmers. By analyzing historical trends, weather forecasts, and soil conditions, AI helps farmers optimize their practices, minimize risks, and maximize productivity.

Sustainability and Resource Management: AI technologies support sustainable farming practices by optimizing resource usage, reducing environmental impact, and promoting regenerative agriculture. By analyzing data on soil health, water usage, and crop rotation, AI helps farmers make eco-friendly decisions [29].

Narrow Application of AI Technologies in Greenhouses:

When focusing specifically on greenhouses, AI technologies offer tailored solutions to optimize crop production in controlled environments. Here are some key aspects of the narrow application of AI technologies in greenhouse farming:

Climate Control: AI systems in greenhouses regulate temperature, humidity, and light levels to create ideal growing conditions for plants. By continuously monitoring and adjusting these parameters, AI ensures optimal growth and yield.

Crop Health Monitoring: AI tools analyze plant health indicators such as leaf color, growth patterns, and disease symptoms to detect issues early on. This proactive monitoring allows for timely interventions to prevent crop losses.

Automated Irrigation: AI-driven irrigation systems in greenhouses deliver precise amounts of water to plants based on real-time data and plant needs. This automated approach conserves water, reduces waste, and promotes efficient water usage.

Pest and Disease Management: AI technologies can identify pests and diseases in greenhouse crops through image recognition and data analysis. By detecting issues promptly, farmers can implement targeted treatments and prevent the spread of infections.

Crop Optimization: AI algorithms optimize planting densities, nutrient delivery, and light exposure in greenhouses to maximize crop yields. By fine-tuning these factors based on data insights, farmers can achieve higher productivity and quality in greenhouse cultivation [30].

2.5.3 Challenges in AI Adoption for Agriculture

The integration of AI into agriculture promises transformative advancements, yet several challenges hinder its widespread adoption. This section explores the key obstacles encountered in implementing AI technologies within agricultural practices.

Perceptions of AI in Farming: AI is often seen as a concept confined to the digital realm, seemingly disconnected from physical farming tasks. This misconception arises from a lack of familiarity with AI tools, particularly among those outside of the technology sector. Consequently, adoption of AI in agriculture has been slow, with many farmers more accustomed to traditional methods and lacking exposure to AI technologies.

Financial Considerations: While AI solutions offer cost-effectiveness in the long run, the initial investment can be prohibitive, particularly for financially-strained farms and agribusinesses. This poses a challenge, especially for small-scale farmers and those in developing countries. However, as technology evolves, upfront costs may decrease, and avenues such as government grants or private investment could help mitigate financial barriers.

Infrastructure and Adoption Process: Agriculture's lack of infrastructure for AI integration lengthens the adoption process. Even farms with existing technology may struggle to progress. Gradual introduction of AI, starting with simpler technologies, can ease this transition. Providers must approach farmers incrementally, gradually introducing additional tools and features as familiarity grows.

Technological Constraints and Data Quality: AI's reliance on diverse, high-quality data presents challenges, particularly in agriculture where data may be scarce. Technological limitations and data constraints require ongoing research and farmer involvement in decision-making. Manual monitoring of AI decisions during early adoption stages is advisable.

Privacy and Security Concerns: Privacy and security issues surrounding AI in agriculture, including cyberattacks and data breaches, remain unaddressed. Lack of regulations adds more concerns, especially in smart farming applications [31].

Despite these challenges, the potential for AI to revolutionize agriculture remains significant. This potential relies on several factors, including the understanding and utilization of temporal data. Recognizing the importance of temporal data in agricultural decision-making opens avenues for enhancing agricultural practices. In light of this, the next section will focus on time series analysis, a pivotal tool for analyzing temporal data and making informed predictions.

2.6 Time Series

In agriculture, especially in controlled environments such as greenhouses, the effective utilization of AI heavily depends on analyzing temporal data. Understanding the dynamics of various parameters over time is crucial for anticipating future conditions and making well-informed decisions. This brings us to the concept of time series analysis, a powerful technique employed to analyze and interpret data points gathered at consecutive time intervals in the following section, we will delve into the core principles of time series analysis and its relevance to agricultural AI, particularly in the context of predicting climate parameters and optimizing greenhouse operations.

2.6.1 Definition

A time series is a sequence of data points gathered at regular time intervals, typically measuring the same variable or attribute of a specific entity or system. Time series data are frequently used to examine trends, patterns, and relationships over time, and can be applied in a wide range of fields, including finance, economics, engineering, and agriculture [32]. The regular time intervals between these data points provide a chronological framework for analysis, enabling further examination and understanding. Through the study of time series data, analysts can uncover insights into how the measured system or entity behaves and evolves over time, and make informed decisions based on these insights.

2.6.2 Components

Time series components refer to the different underlying patterns or structures that can be identified within a time series data. These components help in understanding and modeling the behavior of the time series. The main components typically include:

Trend: Refers to the smooth long-term direction of a time series. Trends can be increasing, decreasing, or flat, indicating the overall direction of the data over time. For example, a monthly sales revenue time series might show an increasing trend over several years.

Seasonality: Refers to patterns of change in a time series within one year which tend to repeat each period. This component is common in data influenced by seasonal factors such as weather, holidays, or business cycles. For example, a time series of monthly retail sales might show a seasonal pattern with higher figures during winter months and lower figures during summer months.

Cyclicity: Refers to the periodic rise and fall of a time series over a period longer than one year. This pattern is often observed in economic data. For example, a time series of employment levels might showcase a cyclical pattern of hiring and layoffs.

Irregularity (Noise): Refers to random variations that follow no regularity in the occurrence pattern. Irregularities can stem from unpredictable factors such as errors, outliers, or measurement

inaccuracies. For example, a time series of daily stock prices might show irregular variations due to unpredictable market forces.

Level: Refers to the baseline or average value around which the data oscillates. It represents the overall magnitude or scale of the time series. For instance, a of monthly temperatures time series might have a level of 20 degrees Celsius, with fluctuations occurring around this baseline due to other components such as trend, seasonality, or cyclicality [33].

“Figure 2.3” clarifies the different components of time series.

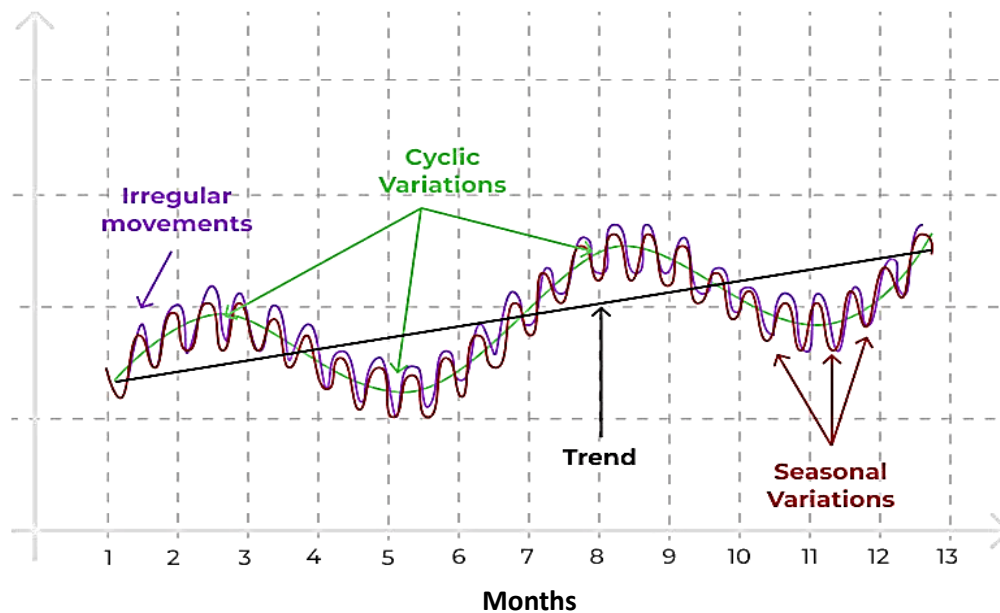


Figure 2.3: Time Series components [33]

2.7 Time Series in Agriculture

Time series analysis is highly valuable in agriculture for improving decision-making and enhancing productivity. By analyzing historical data on crop yields, weather patterns, soil conditions, and pest populations, farmers can make more precise predictions and optimize their farming practices. For instance, time series data can help in forecasting optimal planting and harvesting times, predicting pest outbreaks, and managing irrigation schedules. Moreover, this analysis aids in understanding the long-term impacts of climate change on agricultural productivity and in developing sustainable farming strategies.

2.8 Time Series in Greenhouses

Time series analysis plays a crucial role in greenhouse environments and their optimization for plant cultivation. Greenhouses, designed as controlled environments for plant growth, generate a lot of data that can be analyzed using time series analysis. Within the context of greenhouses,

time series data typically encompass various measurements including temperature, humidity, light intensity, and CO₂ levels, among other factors. These measurements are collected at regular intervals, such as every minute or hour, facilitating continuous monitoring of greenhouse conditions. By analyzing time series data, greenhouse operators can extract valuable insights, identify emerging trends, detect anomalies, and leverage this information to make data-driven decisions regarding the management of the greenhouse environments. This analytical approach not only enhances the operational efficiency but also contributes significantly to the overall health and productivity of the cultivated plants [34].

2.9 Time Series Forecasting

Time series forecasting is the process of predicting future values based on the analysis of past observations within a time series dataset. This predictive analysis often harnesses artificial intelligence techniques, leveraging machine learning algorithms and statistical models. These methods aim to discern patterns and trends in the data, enabling the generation of predictions for future values based on identified patterns.

2.9.1 Forecasting Methods

A forecasting method is a procedure for computing forecasts from present and past values, which can be algorithmic or based on an identified model for the given data. Thus, while the terms "method" and "model" are closely related, they should be kept distinctly separate: a forecasting method refers to the overarching approach or technique used to make predictions, while a forecasting model specifically denotes the mathematical or statistical framework employed within that method to generate forecasts.

Forecasting methods may be broadly classified into three types:

Judgmental Forecasts: These are based on subjective judgment, intuition, insider commercial knowledge, and other relevant information. They often involve expert opinions and qualitative assessments rather than purely quantitative analysis.

Univariate Methods: In univariate forecasting, predictions are made solely based on the historical data of the single variable being forecasted. Common techniques include methods like moving averages, exponential smoothing, and autoregressive integrated moving average (ARIMA) models.

Multivariate Methods: Multivariate forecasting involves using information from multiple time series variables to make predictions. These additional variables, known as predictor or explanatory variables, can help improve the accuracy of forecasts by capturing relationships and dependencies between different factors. Multivariate methods can include techniques like vector autoregression (VAR) models, structural time series models, and machine learning algorithms [35].

2.9.2 Forecasting Models

Following forecasting methods, the focus shifts to specialized neural network architectures adept at processing sequential data, crucial for tasks such as time series prediction.

▪ RNN

RNNs are a type of artificial neural network architecture designed to efficiently process sequential data such as time series by maintaining a memory of past inputs.

RNN vs Traditional Neural Networks

Unlike traditional Feedforward Neural Networks, also known as Multi-Layer Perceptrons (MLPs), which process information linearly without incorporating loops, RNNs have connections that cycle information back through a hidden state H_{t-1} , in conjunction with the network input. This mechanism endows RNNs with the ability to capture temporal dynamics, allowing them to effectively model and forecast sequences of data. As a result, RNNs are extensively applied in diverse fields including language modeling, speech recognition, natural language processing, and time series prediction.

“Figure 2.4” presents a simplified architecture of RNN.

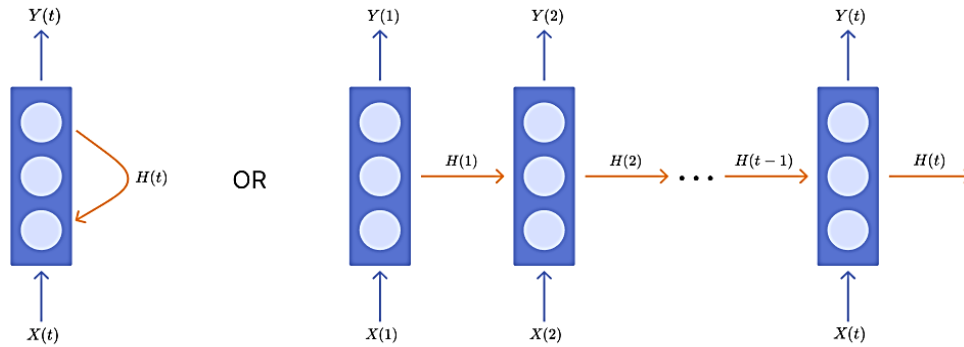


Figure 2.4: RNN architecture

RNNs Workflow

Initialization: At the first time step ($t=1$), you typically initialize the hidden state H_0 to a fixed value or learn it as a parameter.

Input Processing and Hidden State Update: At each time step (t) the network takes the current input X_t and the hidden state from the previous time step H_{t-1} and it calculates a new hidden state using an update equation. This equation typically involves a weighted sum of the current input and the previous hidden state, followed by an activation function:

$$H_t = f(W_{hx}X_t + W_{hh}H_{t-1} + b_h) \quad (1)$$

where W_{hx} and W_{hh} are weight matrices, b_h is a bias term, and f is an activation function.

Output Calculation: The hidden state H_t is then used to calculate the output Y_t at the current time step:

$$Y_t = f(W_{yh}H_t + b_y) \quad (2)$$

where W_{yh} is the weight matrix connecting the hidden state to the output, b_y is a bias term, and f is an activation function.

Feedback to Next Time Step: The hidden state H_t becomes the input to the network at the next time step ($t+1$), and the process repeats. This creates a recurrent loop, allowing the network to capture sequential dependencies.

RNNs Problems

As in most neural networks, it exists key problems in RNNs

After computing the forward pass for the entire sequence in the neural network, a loss function L is calculated based on the predictions $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_t$ and the actual values $y_1, y_2, \dots, y_t$. This loss function quantifies the difference between the predicted and actual values and serves as a measure of the model's performance. It can have different definitions based on the specific problem (e.g. Mean Squared Error, Hinge Loss, Cross Entropy Loss, etc.).

Next, to train the RNN network, it's crucial to compute gradients with respect to the parameters W_{hx} , W_{hh} , and b_h for updating these weights and biases. Backpropagation Through Time (BPTT) is employed for training, which involves propagating the gradients of the loss function L backward through time. This process considers each time step individually, accumulating gradients across time steps. Gradients for each time step are calculated using the chain rule of calculus. Finally, the gradients for the parameters are obtained by multiplying the gradients across time steps.

That is where the problems of RNNs appear

Vanishing Gradients: Vanishing gradients occur as the gradients calculated during backpropagation diminish while propagating backward through time steps. Mathematically, this arises from the repeated multiplication of weights in the gradients, particularly when these weights have values less than 1, causing them to progressively decrease with each time step and eventually vanish. Consequently, earlier time steps in the sequence may have little influence on updating the weights, effectively stopping the contribution of states that occurred far earlier than the current time step towards the current time step, making it challenging for the RNN to capture long-range dependencies.

Exploding Gradients: Exploding gradients occur when the gradients calculated during backpropagation grow uncontrollably as they propagate backward through time steps. This happens due to the repeated multiplication of weights in the gradients, particularly when these weights have

values greater than 1, causing the gradients to exponentially increase with each time step. Consequently, the gradients become so large that they cause instability during training, resulting in large weight updates that may lead to divergence or oscillations in the optimization process. This instability makes it difficult for the RNN to converge to an optimal solution and can hinder its ability to effectively learn from the data [36].

▪ LSTM

Long Short-Term Memory (LSTM) is a variant of RNNs architecture designed to address the vanishing and exploding gradient problems often encountered in traditional RNNs.

LSTM networks consist of specialized memory cells with gating mechanisms that regulate the flow of information, maintaining both hidden and cell states. The hidden state serves as the current memory of the LSTM cell, updated based on the input and the previous hidden state. Meanwhile, the cell state acts as a long-term memory, undergoing updates through operations involving the forget gate, input gate, and output gate. The forget gate determines what information to discard from the cell state, the input gate controls what new information to store, and the output gate decides what information to output. By selectively updating the cell state through these gates, LSTMs can effectively capture and retain long-term dependencies in sequential data.

“Figure 2.5” demonstrates the architecture of an LSTM network.

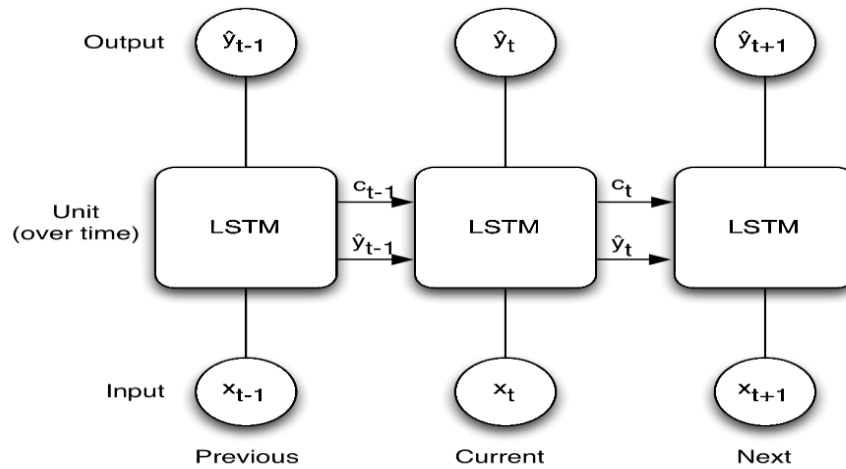


Figure 2.5: LSTM network architecture [37]

LSTM vs RNN

LSTM and RNN are both architectures used for processing sequential data, but they differ in their ability to capture long-term dependencies and mitigate the vanishing gradient problem. While RNNs maintain a hidden state to remember past inputs, they often struggle to retain information over long sequences due to vanishing gradients. In contrast, LSTMs address this limitation by introducing specialized memory cells with gating mechanisms that regulate the flow of information, allowing them to selectively retain or discard information over time. This enables

LSTMs to effectively capture long-term dependencies in sequential data, making them well-suited for tasks such as natural language processing, speech recognition, and time series prediction where understanding context over extended sequences is crucial.

LSTMs Workflow

Input at Time Step t: Receive the input (X_t) at the current time step.

Compute Gates and Candidate Cell State: Calculate the forget gate (f_t), input gate (i_t), and candidate cell state ($\tilde{C}_t$) based on the current input (X_t) and the previous hidden state (H_{t-1}) as follows:

a. **The forget gate:** Decides which information from the cell state to forget or to keep

$$f_t = \sigma(W_{xf}X_t + W_{hf}H_{t-1} + b_f) \quad (3)$$

b. **The input gate:** Determines what new information to store in the cell state

$$i_t = \sigma(W_{xi}X_t + W_{hi}H_{t-1} + b_i) \quad (4)$$

c. **The candidate cell state:** A potential cell state to be added to the current cell state

$$\tilde{C}_t = \tanh(W_{x\tilde{c}}x_t + W_{h\tilde{c}}h_{t-1} + b_{\tilde{c}}) \quad (5)$$

Update Cell State: Use the gates and the candidate cell state to update the cell state (C_t)

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \quad (6)$$

Compute Output Gate: Decides what part of the cell state to output as the hidden state

$$o_t = \sigma(W_{xo}X_t + W_{ho}H_{t-1} + b_o) \quad (7)$$

Compute Hidden State: Compute the final hidden state based on the updated cell state and the output gate

$$H_t = o_t \tanh(C_t) \quad (8)$$

The hidden state H_t is the output of the LSTM cell at time step (t)

Recurrent Connection: Both the hidden state H_t and the cell state C_t are passed to the next time step (t+1) as inputs [37].

“Figure 2.6” illustrates the architecture of an LSTM cell.

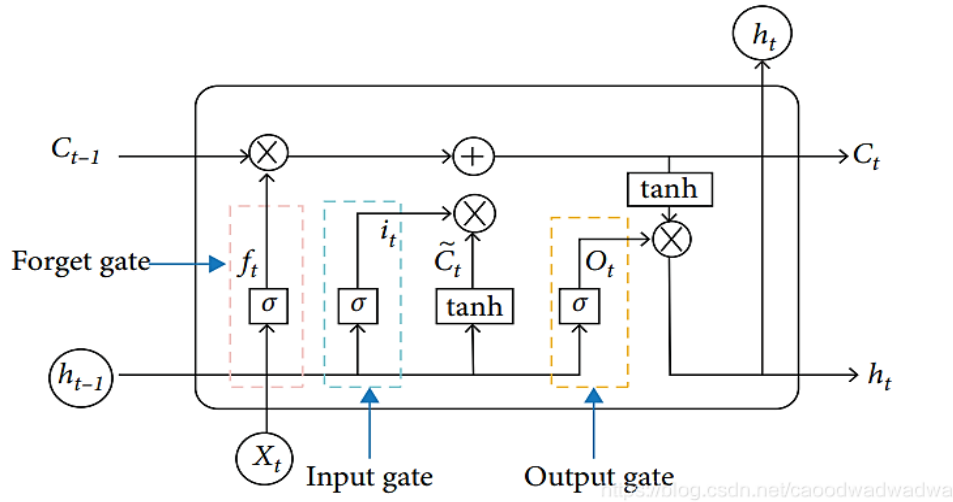


Figure 2.6: An LSTM cell architecture [38]

LSTM Operational Configurations

LSTM networks offer flexibility in how they handle input-output relationships through different operational configurations. These configurations define how the network processes input and generates output, depending on the specific requirements of the task at hand.

One-to-One (1:1): In this configuration, each input corresponds to a single output. It's commonly used in tasks such as classification or regression, where the input and output sequences have a one-to-one mapping.

One-to-Many (1:M): In a one-to-many configuration, a single input generates multiple outputs. This setup is usually used in tasks like image captioning, where a single image input generates a sequence of words as the output caption.

Many-to-One (M:1): Conversely, the many-to-one configuration involves utilizing multiple inputs to generate a single output. This configuration is typical in tasks like sentiment analysis, where multiple words or tokens serve as inputs, and the output is a single sentiment label.

Many-to-Many (M:M): In a many-to-many configuration, both the input and output sequences contain multiple elements. This setup is frequently encountered in tasks like machine translation, where input sequences in one language are translated into output sequences in another language [37].

▪ GRU

The Gated Recurrent Unit (GRU) represents an advanced form of RNNs developed specifically to address fundamental challenges encountered in traditional RNNs, including the problem of vanishing gradients.

GRU vs LSTM vs RNN

GRUs and LSTMs networks are both popular variants of the RNN designed to address the vanishing gradient problem and capture long-term dependencies in sequential data.

GRUs and LSTMs are similar in many respects, while both architectures incorporate gating mechanisms to control the flow of information through the network over time, they differ in complexity. GRUs have a simpler structure with fewer gating mechanisms compared to LSTMs, making them computationally more efficient and easier to train with fewer parameters.

GRU has a simpler memory mechanism compared to LSTM, with only a single hidden state, while LSTM maintains separate cell and hidden states.

GRUs Workflow

Compute the Gates:

a. The reset gate: Determines how much of the previous hidden state H_{t-1} should be forgotten or ignored. It is computed based on the current input X_t and the previous hidden state H_{t-1} :

$$r_t = \sigma(W_r X_t + W_{rh} H_{t-1} + b_r) \quad (9)$$

b. The update Gate: Determines how much of the previous hidden state H_{t-1} to retain and how much of the new candidate state $\tilde{H}_t$ to incorporate. It is computed based on the current input X_t and the previous hidden state H_{t-1} :

$$z_t = \sigma(W_z X_t + W_{zh} H_{t-1} + b_z) \quad (10)$$

Compute the Candidate Hidden State: The candidate hidden state represents the information that could be stored in the current time step. It is computed based on the current input X_t , the reset gate r_t , and the previous hidden state H_{t-1} :

$$\tilde{H}_t = \tanh(W_{hx} X_t + W_{hh} H_{t-1} + b_h) \quad (11)$$

Final Hidden State: The final hidden state represents the updated memory or information at the current time step. It is computed by combining the previous hidden state H_{t-1} and the candidate hidden state $\tilde{H}_t$ using the update gate z_t :

$$H_t = (1 - z_t) \odot H_{t-1} + z_t \odot \tilde{H}_t \quad (12)$$

The symbol $\odot$ typically represents element-wise multiplication.

Output Calculation: Y_t is calculated based on the final hidden state H_t in various ways depending in the task [39].

“Figure 2.7” displays the architecture of a GRU network.

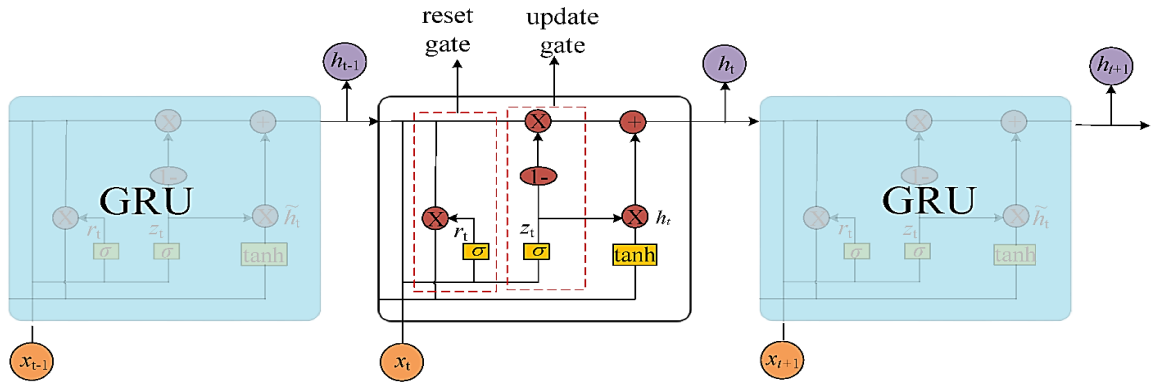


Figure 2.7: GRU network architecture [40]

Algorithm Comparison and Selection

“Table 2.1” summarizes the key strengths and limitations of each forecasting model discussed.

Algorithm	Strengths	Limitations	Greenhouse Application
RNN	Simplicity, good for sequential data	Vulnerable to vanishing gradient issue	Suitable for short-term dependencies
LSTM	Excels at capturing long-term dependencies	More complex architecture	Highly effective for complex tasks and mitigating vanishing gradient problem
GRU	Simplified version of LSTM, faster training	Less powerful than LSTM for long-term dependencies	Balanced approach for moderate complexity

Table 2.1: Algorithms comparison

While each algorithm has its strengths, combining multiple algorithms can leverage the strengths of each, resulting in a more robust solution. For our approach, we have chosen to combine LSTM and RNN to capitalize on the unique advantages of both models. RNNs are well-suited for processing sequential data, making them effective for identifying short-term patterns in greenhouse climate data. In contrast, LSTMs excel at modeling long-term dependencies and overcoming issues like the vanishing gradient problem, making them ideal for capturing trends and patterns over extended periods. By integrating LSTM's long-term memory capabilities with RNN's sequential data processing strengths, we can achieve more accurate and robust forecasting results. This hybrid approach is particularly effective for greenhouse climate forecasting, where both short-term fluctuations and long-term trends are significant.

2.10 Conclusion

In summary, this chapter provides an in-depth exploration of the convergence of AI and agriculture, with a specific emphasis on time series analysis and forecasting. It highlights AI's pivotal role in modern technology and industry, particularly in greenhouse management and agricultural decision-making. By dissecting time series components and AI-driven insights, it illuminates the potential to enhance productivity and resource management in farming.

Looking ahead, the integration of AI and time series analysis holds promise for significant advancements in agricultural systems, facilitating smarter, data-driven approaches to farming. By harnessing the power of AI-driven predictive analytics and optimization algorithms, farmers can adapt to dynamic environmental conditions, optimize resource utilization, and ensure long-term sustainability.

In the following chapter, the transition from theory to practical implementation will be explored, examining how AI and time series analysis solutions are deployed in real-world agricultural settings. Through a case study, the effective deployment of these technologies to address the evolving needs of modern agriculture will be demonstrated, paving the way for a more resilient and sustainable agricultural system.

CHAPTER 3.

GreenSense AI: Architecture, Workflow, and Tools

3.1. Introduction

The integration of advanced technologies into agriculture has opened new frontiers in optimizing resource use, enhancing crop yields, and ensuring sustainable farming practices. Following the exploration of IoT and AI's transformative potential in agriculture, this chapter delves into the practical implementation of a solution tailored for greenhouse climate management.

Greenhouses provide a controlled environment that significantly mitigates external weather challenges. However, maintaining the optimal climate within a greenhouse requires precise monitoring and management of parameters like temperature, humidity, light, and CO2 levels. Traditional climate control methods often lack the consistency and responsiveness needed for modern agriculture.

This chapter outlines the architecture and components of our IoT and AI-powered greenhouse climate management system, covering sensors deployment, communication networks, and AI integration for data analysis and decision-making. By the end of this chapter, readers will gain a comprehensive understanding of how this innovative approach can revolutionize greenhouse farming, leading to higher productivity and more sustainable practices.

3.2 GreenSense AI

3.2.1 The Meaning Behind "GreenSense AI"

“GreenSense AI” is a name that cleverly intertwines agriculture, IOT and AI. Let's the meaning behind it:

- GreenSense: “Green” typically symbolizes agriculture, crops, and the environment. It evokes the idea of nature, growth, and sustainability. “Sense” implies perception, awareness, and understanding through IoT. So, “GreenSense” indicates a perceptive approach suited to agricultural or environmental contexts. Additionally, it subtly resonates with “Greenhouse”, further emphasizing that this solution primarily focusses on greenhouse applications.
- AI: Abbreviating “Artificial Intelligence”. In this context, AI highlights the technological aspect behind the solution, with sophisticated AI algorithms and methodologies.

Bringing these elements together, “GreenSense AI” hints at a solution leveraging AI and IOT to enhance awareness and understanding in agricultural contexts, particularly in the realm of greenhouse management and climate prediction. It implies a system intelligently sensing and interpreting environmental data to optimize agricultural practices, improve crop yields, and promote sustainability.

3.2.2 Solution Workflow

Our solution, GreenSense AI, seamlessly integrates IoT and AI technologies to revolutionize greenhouse climate management. This innovative approach marks a significant advancement in agriculture, where cutting-edge technology converges with sustainable practices to optimize greenhouse climate conditions.

Monitoring Crucial Climate Parameters: At the core of GreenSense AI is a network of IoT devices distributed throughout the greenhouse. These sensor-equipped devices continuously monitor essential climate parameters in real-time, which are temperature, humidity, soil moisture, raindrops, CO₂ levels, and light intensity. Through continuous data collection, GreenSense AI provides a comprehensive understanding of the greenhouse environment.

AI-Powered Predictive Analytics: The raw data collected by the IoT sensors undergo thorough analysis by advanced AI algorithms. Leveraging machine learning and predictive analytics, including Recurrent Neural Networks, the system forecasts future climate conditions with remarkable accuracy.

Precision Climate Control and Optimization: Using predictive insights, GreenSense AI enables greenhouse operators to precisely adjust their climate control systems. The AI-generated recommendations provide actionable insights into optimal environmental conditions for various crops at different growth stages. By implementing these recommendations, farmers can create an environment that promote optimal plant growth while minimizing energy consumption and resource use.

User-Friendly Web Application: GreenSense AI extends its functionality beyond the confines of the greenhouse through a user-friendly web application. This intuitive interface allows farmers to effortlessly access and control their greenhouse environment from any network-connected device.

Continuous Improvement and Adaptation: The capabilities of GreenSense AI are not static but continually evolving. The system archives historical data for analysis and visualization, while AI algorithms are regularly updated and refined based on new data insights. This ensures ongoing optimization and adaptability to changing environmental conditions.

Empowering Sustainable Agriculture: In summary, GreenSense AI represents a paradigm shift in greenhouse climate management, bridging the gap between traditional farming practices with advanced technology. By harnessing the power of IoT and AI, GreenSense AI empowers farmers to cultivate crops more efficiently, sustainably, and profitably, introducing a new era of agriculture that prioritizes innovation, productivity, and environmental sustainability.

3.3 Hardware Part

Within the hardware implementation of the project, several components are pivotal. This section discusses the selection of essential hardware elements required for the solution.

3.3.1 Employed Microcontroller

One of the crucial steps in the solution implementation is selecting the appropriate microcontroller. The choice should be based on several criteria, including:

- The number of analog pins on the board
- Inter-Integrated Circuit (I2C) compatibility
- Random Access Memory (RAM) size
- Electrically Erasable Programmable Read-Only Memory (EEPROM) memory (for saving data in case of a power outage)
- Price and market availability
- Community support and resource availability

“Table 3.1” represents a comparative study of some microcontrollers available on the market.

Type of Microcontroller	ESP32	ESP8266	STM32
Flash Rom Max	16MB	16MB	2MB
RAM	8MB	160KB	256KB
Fmax	240Mhz	80Mhz	180Mhz
Analog Pins	12-16	1	16-60
E/S	36	16	More then 40
IOT projects	Very numerous	Very numerous	Growing, focus on industrial use
Community Size	Large and active	Large and active	Moderate
Average Price	5\$	3\$	10\$

Table 3.1: Microcontrollers comparison [41], [42]

These are some of the reasons why the ESP32 is an excellent choice for GreenSense AI. It offers greater memory capacity and more analog pins, allowing for the management of more complex projects. Consequently, using the ESP32 enables additional features and expands the solution's capabilities to meet specific needs.

ESP32 Overview

Stands out for its versatility, cost-effectiveness, robustness, and energy efficiency, the ESP32 is a System on a Chip (SoC) general-purpose microcontroller. It boasts a dual-core processor, extensive peripherals, ample memory capacity, and integrated Wi-Fi and Bluetooth connectivity, setting it apart from other microcontrollers with its superior power efficiency and flexibility.

Featuring a smaller footprint than conventional Arduino boards, the ESP32 features multifunctional pins, including 16 PWM channels and 12 digital and analog pins. While it's compatible with the Arduino IDE after initial setup, it also supports alternative programming languages like Python. However, for most users, the Arduino IDE remains the preferred and most accessible option [43].

“Figure 3.1” offers a visual representation of the ESP32 board, providing some descriptive details.

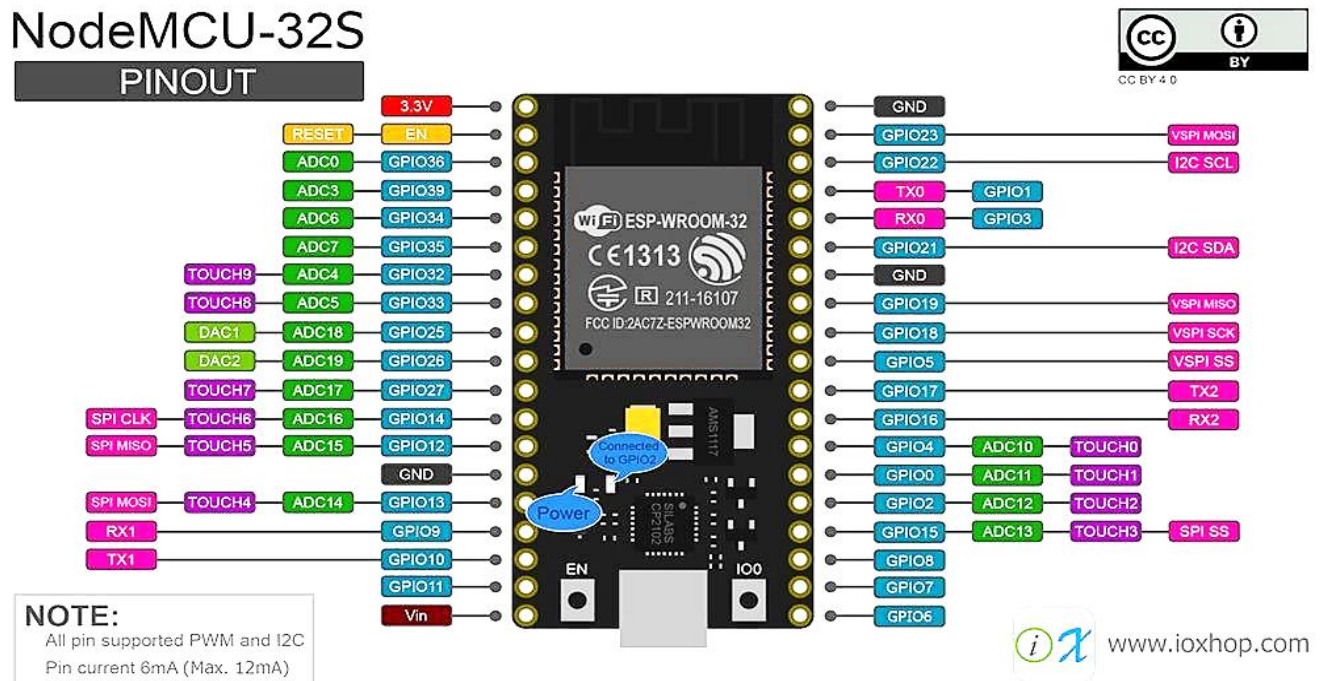


Figure 3.1: Description diagram of ESP32 Board [44]

3.3.2 Raspberry PI 4

The Raspberry Pi is a small, affordable, and powerful single-board computer developed by the Raspberry Pi Foundation. It has revolutionized the world of computing, making it accessible to a wide range of users. Our model, the Raspberry Pi 4, released in 2019, offers improved performance, enhanced connectivity, and increased flexibility compared to previous versions, making it ideal for IoT projects, including ours.

The Raspberry Pi is a series of small, affordable, single-board computers developed by the Raspberry Pi Foundation, aimed at promoting computer science education and facilitating hobbyist and professional projects.

Strategic Use of Raspberry Pi

GreenSense AI aims to integrate IoT and AI technologies effectively. The Raspberry Pi 4, equipped with a 1.5GHz quad-core Advanced RISC Machines (ARM) Cortex-A72 Central Processing Unit (CPU), provides sufficient processing power to execute various applications, especially AI algorithms. Its built-in Wi-Fi and Bluetooth connectivity, combined with its compact size, allows for seamless integration within the greenhouse and wireless access [45].

The Raspberry Pi supports various operating systems, including Raspbian, which was designed specifically for it. Furthermore, its extensive and active developer community provides abundant resources, simplifying development compared to other single-board computers (SBCs). Additionally, the Raspberry Pi is a cost-effective SBC, making it an ideal choice for the project due to its performance and capabilities.

3.3.3 Sensors

In various domains including industry, scientific research, services, and leisure activities, the need to monitor different physical parameters such as temperature, position, speed, and brightness, arises. At the core of this process lies the sensor, a fundamental component responsible for detecting these physical quantities [46].

Definition

A sensor is a device that collects data by transforming a physical quantity into another form, typically electrical. This transformed quantity, which represents the measured parameter, is then employed for measurement or control purposes [46].

Classification of sensors

Energy input: Based on how sensors derive their operating energy, are classified into:

- Passive sensors: Operate autonomously without external energy sources (e.g., thermistors, mercury thermometers). Modeled by impedance, they reflect variations in physical phenomena through impedance changes.
- Active sensors: Necessitate an external power source for operation (e.g., strain gauge, gyroscopes). They are modeled by generators that generate current or voltage proportional to the intensity of the measured physical phenomenon.

Output type: In terms of output, sensors can also be classified into:

- Analog: Providing continuous information (electrical, visual, etc.) varying within defined limits.
- Digital: Delivering data as binary code (with defined bits) or a pulse train (with specific pulses or frequency).
- Logical: Binary sensors (All or Nothing) output only two states, typically represented as 0 and 1 [46].

Employed Sensors

Temperature and Humidity Sensor

The DHT22 sensor is an advanced solution designed for precise measurement of both humidity and temperature within the surrounding environment. Integrating a capacitive humidity sensor with a thermistor, it ensures accuracy. This sensor finds its application in various domains including home automation, environmental monitoring, and weather data tracking [47].

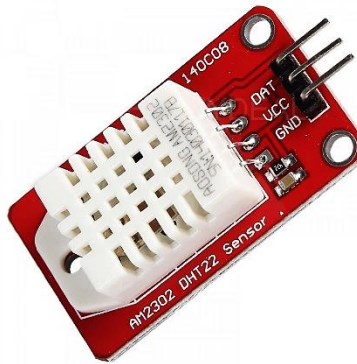


Figure 3.2: DHT22 sensor

This sensor features three pins, each serving a specific purpose:

- VCC: Functions as the power supply pin (accepting either 5 volts or 3.3 volts).
- DAT: Dedicated communication pin.
- GND: Serves as the sensor ground (GND).

Soil Moisture Sensor

The SparkFun Soil Moisture Sensor with Screw Terminals is a user-friendly, opto-isolated board designed to measure moisture levels in soil and similar materials. It features two large exposed pads that act as probes and function as a variable resistor. As soil moisture increases,

conductivity between the pads increases, resulting in lower resistance and a higher external signal. This version includes a pre-soldered 3-pin screw terminal for easy connection and installation [48].

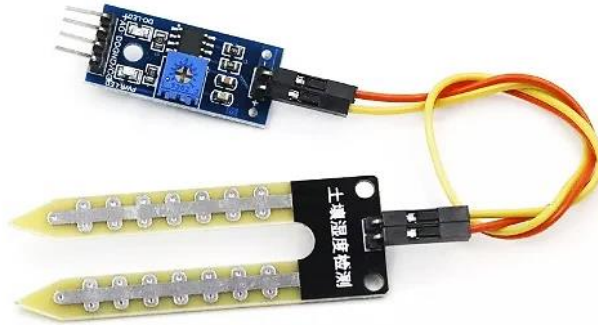


Figure 3.3: SparkFun sensor

This sensor has three terminals, each with a specific function:

- Power supply pin: Connects to the 3.3V pin of the ESP32.
- GND terminal: Connects to ground.
- SIG or A1 terminal: Connects to an analog input of the ESP32.

Light Intensity Sensor

The TSL2561 light intensity sensor is an electronic component with a digital output, designed to measure ambient light intensity with high precision and low power consumption. It uses light detection technology based on dual photodiodes and an integrated analog-to-digital conversion circuit to provide accurate light level measurements [49].

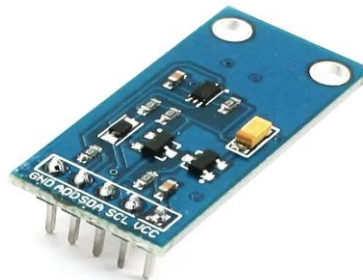


Figure 3.4: TSL2561 sensor

This sensor has six terminals, each serving a specific function:

- VIN: The module's power supply, which typically ranges from 3.0V to 5.0V.
- GND: Module ground, connected to the circuit's ground.

- SCL: Serial clock line, providing a clock pulse for I2C communication.
- SDA: Serial data line, used to transfer data via I2C communication.
- INT: Interrupt output, which can signal the microcontroller under certain conditions.
- ADDR: Device address pin, used to select the I2C address when multiple modules are connected.

CO2 Level Sensor

The CCS811 is a sensor designed to measure CO2 levels in the environment. It features an internal algorithm for calculating equivalent carbon dioxide (eCO2) and total volatile organic compounds (TVOC) concentrations based on its measurements. Operating via I2C communication, the sensor typically requires a power supply voltage between 1.8V and 3.6V. It offers various modes of operation, including sleep mode, and provides accurate measurements for environmental monitoring applications [50].

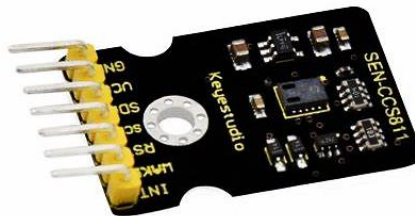


Figure 3.5: CCS811 sensor

This sensor has six terminals, each serving a specific function:

- Vin: Power supply input, typically ranging from 1.8V to 3.6V.
- GND: Ground connection.
- SCL: Serial clock line for I2C communication.
- SDA: Serial data line for I2C communication.
- WAKE/nWAKE: These pins serve as the wake-up function for the sensor from sleep mode.

Rain Drops level Sensor

A raindrops level sensor is a specialized device engineered for accurate measurement of water levels in different environments. Utilizing conductive traces or probes, it detects water presence at varying levels. Upon contact with water, these probes form a circuit, generating a signal readable by a microcontroller like the ESP32. This signal is then utilized for various IoT

applications. Typically operating between 3.3V and 5V, these sensors offer high accuracy, often within ± 1 cm, in detecting water levels [51].



Figure 3.6: Rain Drops level sensor

This sensor has three terminals, each serving a specific function:

- VCC: Connected to a power supply voltage ranging from 3.0V to 5.0V.
- GND: Connected to the ground of the power supply.
- AOUT: Connected to the analog input/output (I/O) of the microcontroller unit (MCU) for analog output.

3.3.4 Actuators

Across numerous fields, including manufacturing and services automation, the management of physical parameters such as motion, force, and positioning is consistently in demand. At the heart of these operations are actuators, which facilitate the execution and control of these tasks. Actuators are indispensable in a variety of applications, from powering robotics and industrial equipment to enhancing consumer electronics and smart home technologies, ensuring precise and efficient functionality [52].

Definition

An actuator is a device that converts energy into physical motion, typically used to control mechanisms or systems. Actuators are essential components in automation and control systems, enabling machines and devices to perform tasks such as moving, opening, or adjusting. They can be powered by various energy sources depending on the requirements of the application [52].

Employed Actuators

Lamp

A plant growth lamp, operating at 12 volts, is specifically designed to provide optimal lighting conditions for photosynthesis in plants. These lamps play a vital role in environments with limited natural light, such as greenhouses, indoor farms, or hydroponic gardens. By delivering the

precise amount and quality of light necessary for photosynthesis, they boost plant growth and health, ensuring their vitality and productivity.



Figure 3.7: Employed lamp

Fan

The fan is a crucial device for managing temperature, humidity, and CO₂ levels within greenhouses. Efficient air circulation from the fan helps prevent hotspots, controls excess humidity, and evenly distributes CO₂, all critical for optimal plant growth. This balanced environment supports robust photosynthesis, resulting in healthier and more productive plants. Whether in greenhouses, indoor farms, or hydroponic systems, we demonstrate with a 12V fan that can be replaced with a larger fan by adjusting the energy source accordingly, ensuring sustained effectiveness in maintaining ideal growing conditions.



Figure 3.8: Employed fan

3.4 Software Part

Building upon the hardware infrastructure, the software aspects in GreenSense AI constitute the digital backbone that enables seamless integration, management, and analysis of data within the greenhouse environment. These tools are essential for optimizing development efficiency, enabling real-time monitoring, and supporting informed decision-making in agricultural management. Let's explore these essential aspects further.

VNC Viewer

Virtual Network Computing (VNC) Viewer is a software application that enables users to remotely access and control a computer or device's graphical desktop interface over a network. It allows users to interact with the remote system as if they were physically present, providing capabilities such as viewing and interacting with graphical user interfaces, running applications, and managing systems from a distance. VNC Viewer supports remote collaboration, troubleshooting, and system administration across distributed computing environments [53].

In GreenSense AI, VNC Viewer serves as a tool for remotely accessing the graphical user interfaces (GUIs) of Raspberry Pi unit that host the entirety of our project's system. This capability allows us to remotely adjust configurations of the system on the Raspberry Pi.

Mosquitto Broker

Mosquitto Broker is an open-source message broker that implements the MQTT (Message Queuing Telemetry Transport) protocol, serving as a central hub for communication in IoT applications. It enables efficient and reliable messaging between sensors, actuators, and other components, facilitating real-time data exchange. Devices within the greenhouse environment can publish messages to specific topics and subscribe to receive relevant messages [54].

In GreenSense AI, Mosquitto Broker plays a pivotal role in facilitating communication and data exchange among various system components within the greenhouse environment. It enables seamless interaction between sensors, actuators, and other devices, allowing for monitoring of environmental conditions, transmitting data efficiently, and executing automated actions promptly based on sensor readings. By serving as a reliable messaging backbone, Mosquitto Broker contributes to the robustness and effectiveness of the GreenSense AI system, ensuring smooth operation and optimal management of greenhouse resources.

Arduino IDE Software

The Arduino Integrated Development Environment (IDE) is a free, open-source cross-platform software tool designed for programming Arduino microcontrollers. It provides an intuitive interface for writing, compiling, and uploading customized code tailored to control various system components, including sensors, actuators, and other associated boards [55].

In the GreenSense AI project, Arduino IDE plays a pivotal role in developing and managing the code executed on ESP32 microcontroller. These microcontrollers are often employed to interface with sensors and actuators within the greenhouse environment. Using Arduino IDE, developers can write and upload code to coordinate these components, enabling operations such as data collection from sensors and automation of climate control tasks.

Visual Studio Code

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It supports editing, debugging, and managing code across various programming languages and platforms. With an intuitive interface, robust debugging tools, built-in Git integration, and a vast library of extensions, VS Code is a versatile and powerful tool for developers [56].

In GreenSense AI, Visual Studio Code serves as an essential development environment for writing and managing project codes. Its powerful features boost productivity and facilitate collaboration among team members, streamlining the development process and enabling developers to write high-quality code efficiently.

GitHub

GitHub is a web-based platform for version control and collaborative software development, built on Git, a distributed version control system. It provides developers with a space to host and review code, manage projects, and collaborate efficiently. GitHub key features include code repositories, issue tracking, project management tools, and continuous integration/continuous deployment (CI/CD) workflows. With its extensive integrations and user-friendly interface, GitHub is an essential tool for modern software development [57].

In GreenSense AI, GitHub is used to manage and version control the project's codebase. It enables seamless collaboration among team members, allowing them to track changes, merge code contributions, and resolve conflicts efficiently. This ensures the project is developed and maintained effectively.

MySQL

MySQL is an open-source relational database management system (RDBMS) that provides a powerful platform for storing, managing, and retrieving structured data. It is widely used across diverse applications and industries due to its reliability, scalability, and robust features. Operating on a client-server architecture, MySQL allows clients to interact with the MySQL server to execute database operations such as querying, updating, and data management. By Supporting SQL (Structured Query Language), MySQL enables the definition and manipulation of databases, while also offering advanced functionalities like transactions, indexing, and replication to ensure efficient data handling and high availability [58].

Within GreenSense AI, MySQL plays a crucial role in storing and managing data related to greenhouse climate monitoring and control. This includes recording sensor readings and maintaining comprehensive historical records. Leveraging MySQL's capabilities, this data can be easily queried, analyzed by AI algorithms to forecast greenhouse climate conditions, and visualized alongside predictions. Such insights empower the optimization of greenhouse conditions and facilitate informed decision-making for agricultural management.

Node-Red

Node-RED is an open-source visual programming tool crafted for creating event-driven applications. It provides a browser-based flow editor where users can connect diverse nodes, representing functions or devices, to construct flows. Node-RED is particularly popular in the IoT space for its ability to swiftly integrate hardware devices, Application Programming Interfaces (APIs), and online services into automation workflows. With its flow-based programming approach, Node-RED simplifies the development process, making it accessible to users with varying technical backgrounds [59].

In GreenSense AI, Node-RED serves as a powerful tool for constructing and deploying automation workflows tailored to greenhouse monitoring and control. Through its user-friendly visual interface, creating flows is effortless, enabling the collection of sensor data, analysis of environmental conditions, and execution of actions like climate control adjustments or alert notifications. Node-RED's extensive library of pre-built nodes and integrations further simplifies connectivity with various devices and services, facilitating seamless integration with existing systems.

Python

Python is a high-level, interpreted programming language known for its simplicity and readability. Its dynamic typing and binding make it an ideal choice for rapid application development and scripting. Supporting multiple programming paradigms, including procedural, object-oriented, and functional programming, Python comes with a comprehensive standard library that provides modules and packages for a wide range of tasks. This versatility and ease of use have made Python a popular choice among developers across various domains [60].

In the context of GreenSense AI, Python serves as a versatile programming language for developing and implementing various aspects of the AI system. It is used for tasks such as data preprocessing, deep learning model training and inference, real-time data analysis, and system integration. Python's extensive libraries and frameworks, such as NumPy, pandas, TensorFlow, and scikit-learn, provide powerful tools for data manipulation, statistical analysis, and building sophisticated deep learning algorithms.

Jupyter Notebook

Jupyter Notebook is an open-source web application that enables users to create and share documents combining live code, equations, visualizations, and narrative text. It supports various programming languages, with a strong emphasis on Python, and is widely used for data analysis, scientific research, machine learning, and academic projects. Jupyter Notebook provides an interactive environment that integrates code execution, rich text, and multimedia, making it an invaluable tool for exploratory data analysis, visualization, and iterative development [61].

In the context of GreenSense AI, Jupyter Notebook serves as an essential tool for developing, documenting, and sharing project-related code. It provides an interactive computing environment where developers can collaboratively work on documents containing live code, equations, visualizations, and narrative explanations. This functionality is particularly valuable for tasks such as data exploration, visualization, and analysis, as well as for prototyping and debugging AI models. By integrating code execution with rich text, Jupyter Notebooks facilitate collaboration and communication among team members, enabling efficient development.

3.5 Conclusion

In conclusion, this chapter explores how IoT and AI technologies integrated via GreenSense AI revolutionize greenhouse climate management. Utilizing a network of sensors and actuators powered by microcontrollers like the ESP32, alongside a Raspberry Pi 4, GreenSense AI enables precise monitoring and control of key environmental parameters such as temperature, humidity, light intensity, CO2 levels, and soil moisture.

Complementing this hardware infrastructure, a robust suite of software tools plays a pivotal role in the functionality and management of GreenSense AI. Tools such as VNC Viewer, GitHub, MySQL, and Jupyter Notebook facilitate data collection, analysis, and real-time monitoring, empowering farmers with actionable AI-driven insights to make informed decisions that enhance productivity, reduce resource consumption, and promote sustainable agricultural practices.

Overall, GreenSense AI represents a significant leap in agriculture by seamlessly integrating cutting-edge technology with traditional farming practices. It not only optimizes greenhouse climate management but also sets a new standard for eco-friendly farming solutions, addressing global food security and sustainability challenges.

The next chapter delves into the practical implementation of GreenSense AI solution, detailing setup process, hardware-software integration, and deployment methodology. Additionally, it presents the results obtained from the implementation, showcasing GreenSense AI's impact on greenhouse climate management improvement. This will provide a comprehensive view of how the approach performs in real-world scenarios and its potential to revolutionize modern agriculture.

CHAPTER 4.

GreenSense AI: Implementation and Results

4.1 Introduction

After delving into the theoretical foundation of GreenSense AI, examining the interplay of IoT and AI, and outlining the project's conception and requirements in previous chapters, we now turn to the practical execution. This chapter focuses on the journey from design to implementation, detailing the stages involved in bringing GreenSense AI to life.

GreenSense AI's development can be divided into two primary phases: data acquisition and climate prediction. Each phase encompasses several smaller, critical steps, which we will explore in depth. From the initial crafting to the final results, this chapter provides a comprehensive overview of the processes and methodologies employed, shedding light on how theory transforms into practice.

4.2 Data Acquisition

This part is primarily driven by an IoT architecture, equipped with various sensors and microcontrollers, to monitor and capture key climate factors inside and outside the greenhouse.

The data acquisition process involves several key steps. It begins with sensor measurement, where each sensor records a specific environmental factor. Following this, the ESP32 microcontroller collects these measurements. The data is then transmitted to a Raspberry Pi, which serves as the central hub of the system. The Raspberry Pi stores the data in a database and also publishes it to an MQTT broker. The Node-RED service on the Raspberry Pi subscribes to these data streams, processing the information and preparing it for visualization. Each of these steps will be explored in detail in this section.

"Figure 4.1" simplifies the data acquisition process with a clear diagram.

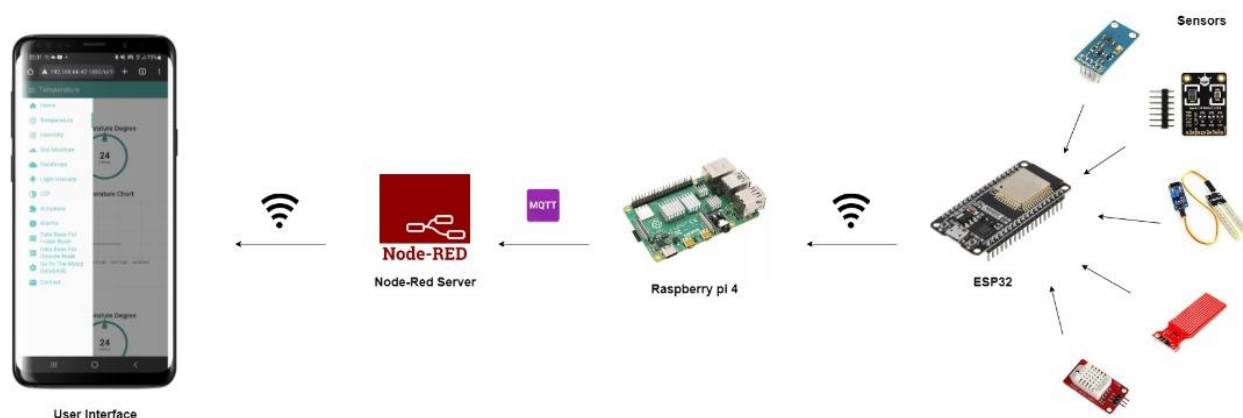


Figure 4.1: Data Acquisition process

4.2.1 Sensor Measurement

The process begins with various sensors strategically placed on both interior and exterior nodes of the greenhouse. These sensors monitor essential environmental factors such as temperature, humidity, soil moisture, and light intensity. Each sensor is carefully selected for its accuracy and reliability in capturing real-time data.

Temperature and Humidity Monitoring

The first two pivotal factors in the greenhouse climate are temperature and humidity. Maintaining stability in these factors is paramount, as fluctuations can jeopardize crops. These parameters are measured at both the exterior and interior nodes of the greenhouse, as conditions in both environments influence the internal climate.

The DHT22 sensor is responsible for these measurements. The setup in “Figure 4.2” illustrates how this sensor is deployed at each measurement node.

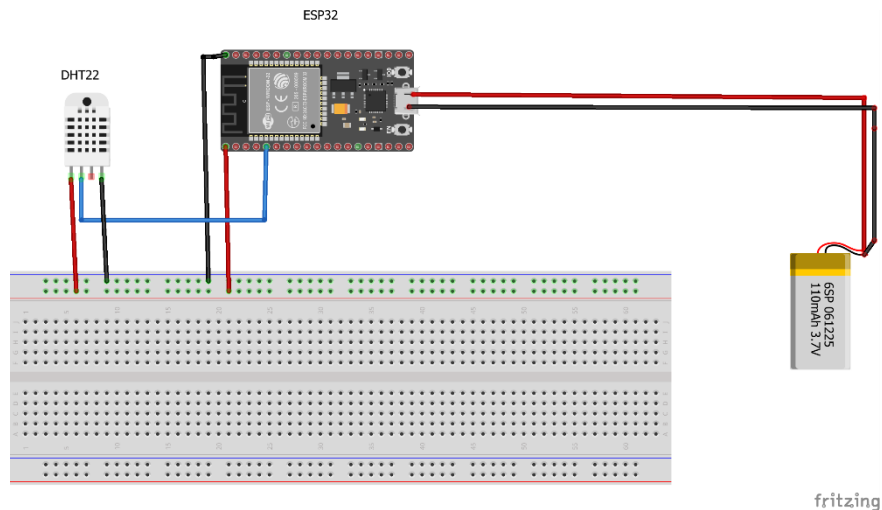


Figure 4.2: DHT22 sensor deployment

Soil Moisture Monitoring

Another crucial factor in greenhouse climate that requires monitoring is soil moisture. The primary goal is to ensure optimal water availability for crops, thereby enhancing agricultural productivity and mitigating the risk of water scarcity. Soil moisture is measured exclusively at the interior node, as research indicates that external soil moisture levels do not significantly affect the internal climate of the greenhouse.

The measurement is conducted using the SparkFun sensor. The deployment of this sensor at the interior node is illustrated in “Figure 4.3”.

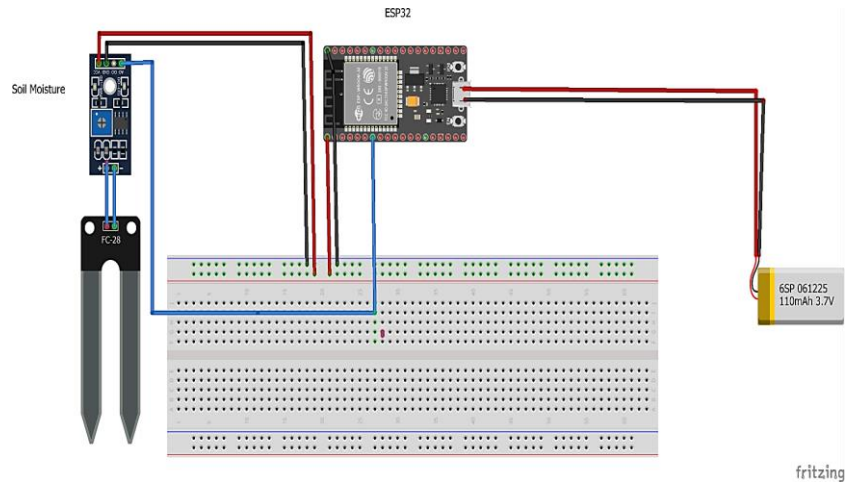


Figure 4.3: SparkFun sensor deployment

Light Intensity Monitoring

Light intensity is a critical aspect of greenhouse conditions that requires careful monitoring. Ensuring adequate light levels is essential for photosynthesis, directly optimizing plant growth and productivity, and efficiently managing energy use. Light intensity is measured at both the interior and exterior nodes because both indoor and outdoor light levels influence the greenhouse's climate.

The measurement of light intensity is carried out using the TSL2561 sensor, as depicted in “Figure 4.4”, illustrating its deployment at both node for accurate data acquisition.

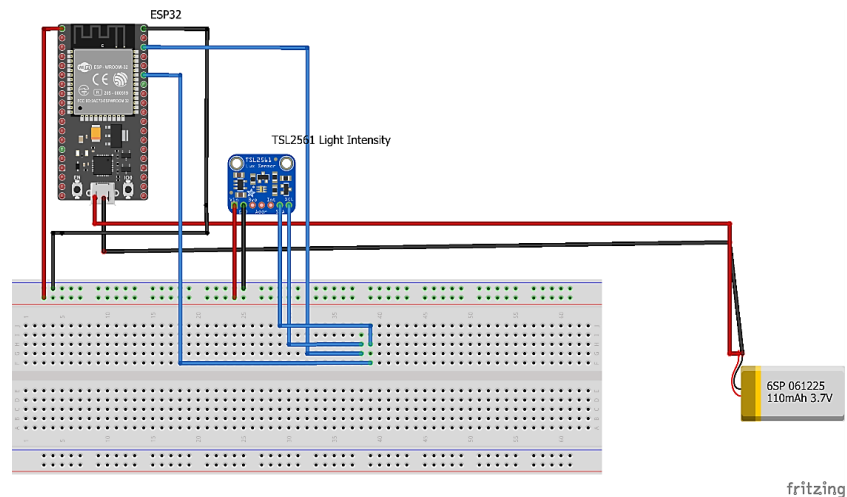


Figure 4.4: TSL2561 sensor deployment

CO2 Level Monitoring

Monitoring CO2 levels is crucial in greenhouse management. Maintaining the right CO2 levels is essential for maximizing productivity by supporting efficient photosynthesis. Research has shown that CO2 levels outside the greenhouse remain relatively constant and have minimal impact on the interior climate. Therefore, CO2 levels are monitored exclusively at the interior node.

The CCS811 sensor is employed to provide precise, real-time measurements of CO2 levels, as showcased in “Figure 4.5”, highlighting its deployment within the inside node.

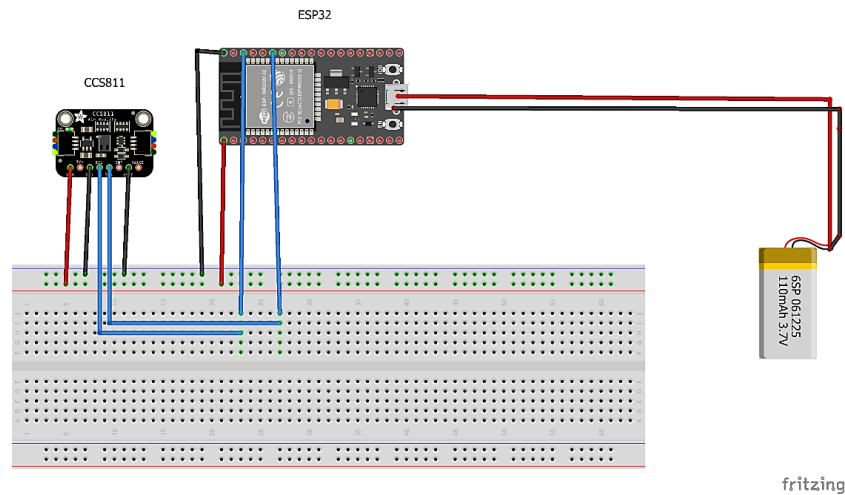


Figure 4.5: CCS811 sensor deployment

Rain Drops Level Monitoring

Monitoring raindrop levels in a greenhouse setting is essential for effective water management and maintaining optimal growing conditions. It helps control irrigation schedules, preventing both overwatering or underwatering of plants, and ensures that the greenhouse environment remains conducive to plant growth. Since greenhouses are enclosed environments, measurements of raindrop levels are typically limited to sensors positioned outside the greenhouse.

The raindrop level sensor is strategically deployed outside the greenhouse to provide accurate, real-time measurements, as shown in “Figure 4.6”.

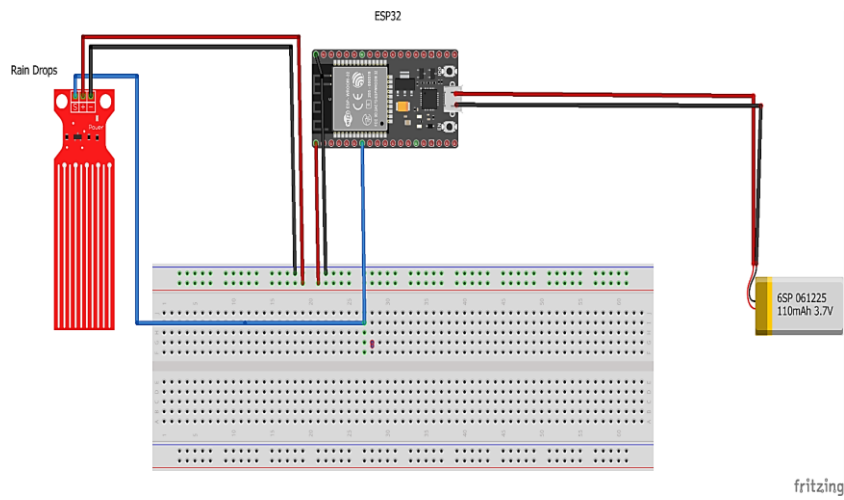


Figure 4.6: Rain Drops sensor deployment

4.2.2 ESP32 Microcontroller Data Collection and Transmission

In the GreenSense AI system, the ESP32 microcontroller plays a pivotal role in efficiently collecting data from various environmental sensors deployed within and around the greenhouse. Each ESP32 is powered by a power bank and programmed using Arduino IDE to interface with specific sensors strategically positioned to monitor key climate parameters. Leveraging its built-in capabilities, the ESP32 continuously reads sensor outputs and processes them as required. It then utilizes Wi-Fi (TCP/IP) wireless communication to transmit the collected data in JavaScript Object Notation (JSON) format to a centralized processing unit, typically the Raspberry Pi. This seamless data transmission is essential for real-time monitoring and analysis of environmental conditions. With robust connectivity, low-power consumption, and efficient data handling capabilities, the ESP32 is an ideal choice for enhancing environmental monitoring within the GreenSense AI system.

4.2.3 Raspberry Pi Data Management

In the implementation of GreenSense AI, the Raspberry Pi assumes a critical role in the data acquisition phase as the central data processing hub within the IoT architecture. It serves as the primary endpoint for data collected. Upon receiving data from the ESP32, the Raspberry Pi orchestrates a series of essential tasks to ensure effective data management and system operation.

The Raspberry Pi performs two primary functions during this phase: aggregating and organizing collected environmental data, and publishing it via MQTT. It aggregates the data and

structures it into two databases tailored for subsequent AI processes: a comprehensive database containing all collected data, and a specialized partial database containing recent collected data, both saved in Comma-Separated Values (CSV) format. Next, the Raspberry Pi publishes the aggregated data to designated topics on an MQTT broker. This MQTT communication protocol ensures efficient and reliable distribution of data across the network, facilitating seamless integration with other system components, such as data visualization tools and control mechanisms.

Moreover, the Raspberry Pi's flexibility and support for diverse IoT protocols make it adaptable to evolving system requirements, enabling scalability as the GreenSense AI system expands. These capabilities are fundamental for ensuring real-time monitoring, analysis, and management of environmental conditions within the greenhouse, thereby supporting informed agricultural practices and enhancing productivity.

4.2.4 Node-RED Data Handling and Visualization

Node-RED serves as a powerful visual programming tool for managing data flows and visualizing information within the GreenSense AI system. The Node-RED service on the Raspberry Pi begins by subscribing to MQTT topics where sensor data, previously published by the Raspberry Pi, is distributed in real-time. Upon receiving this data from the MQTT broker, Node-RED initiates data processing via predefined workflows using a visual programming approach. This enables parsing, filtering, and transforming the data as necessary. The processed data is then utilized to generate dynamic dashboards, charts, and graphs that provide stakeholders with clear insights into current and historical environmental conditions within the greenhouse. These visualizations are crucial for monitoring and optimizing the system's performance.

“Figure 4.7”, “Figure 4.8”, and “Figure 4.9” illustrate the interior node, exterior node, and recommendation flows within Node-RED, each corresponding to different tabs of the dashboard.

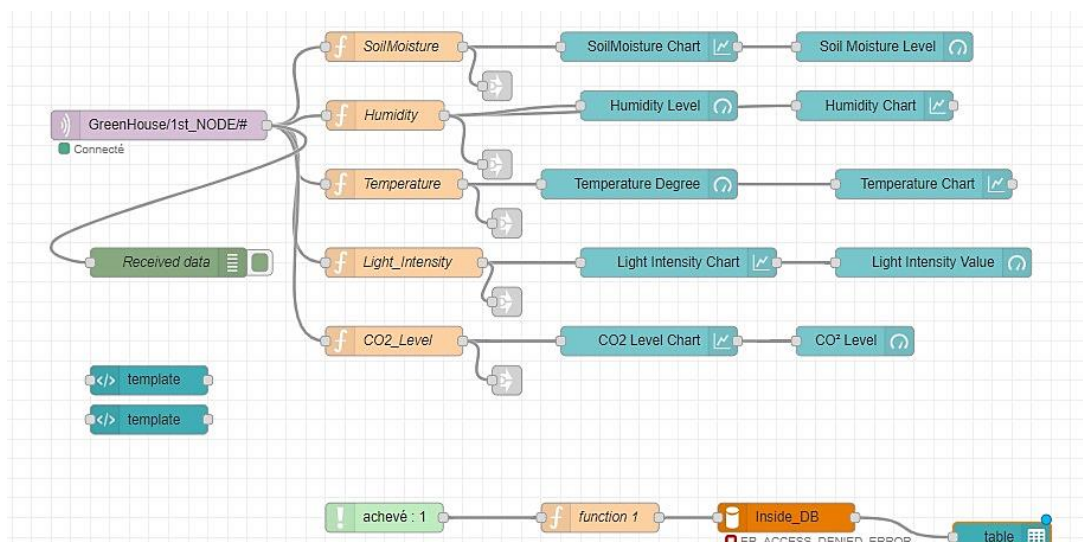


Figure 4.7: Interior node flow

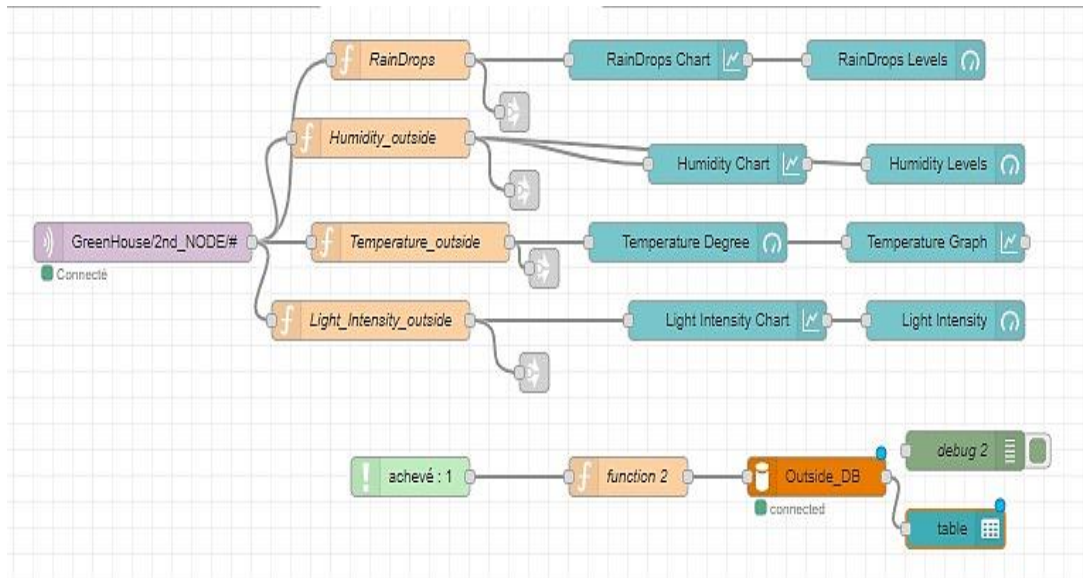


Figure 4.8: Exterior node flow

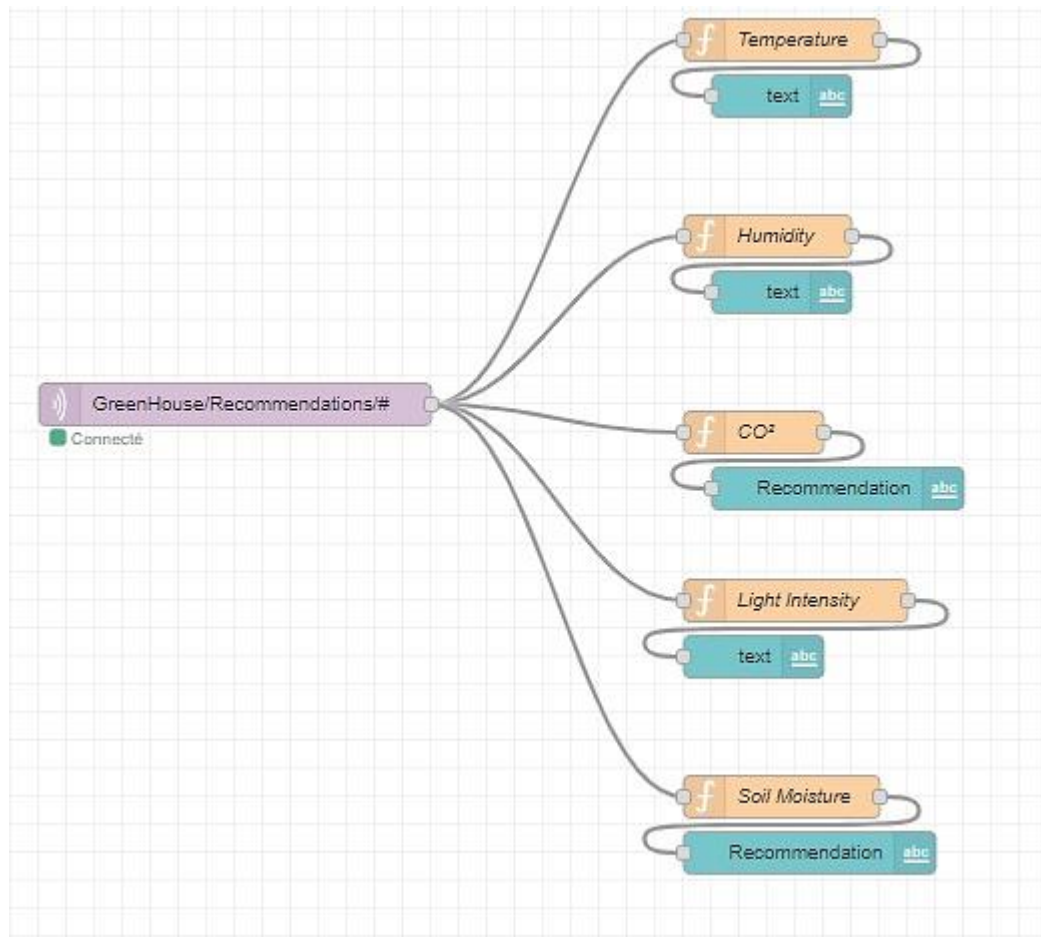


Figure 4.9: Recommendations flow

To clarify, both the Node-RED service and the MQTT broker are located on the Raspberry Pi connected to the ESP32 microcontrollers. This setup is used because it is a prototype involving a single greenhouse. In larger scale real-world scenario with multiple greenhouses, each greenhouse would have its own set of ESP32 microcontrollers and a Raspberry Pi for processing. These Raspberry Pis from different greenhouses would then connect to a central Raspberry Pi via the MQTT protocol. This central Raspberry Pi acts as the MQTT broker, receiving data published by the Raspberry Pis from individual greenhouses. It also hosts the Node-RED service responsible for the data processing mentioned earlier.

4.3 Climate Prediction

The climate prediction component of GreenSense AI utilizes advanced AI techniques to forecast environmental conditions within the greenhouse. The process begins by processing data prepared by the Raspberry Pi, ensuring it is clean and structured for optimal use. Next, the AI model is constructed, involving training and evaluation on extensive climate datasets to determine the best architecture and hyperparameters. Once the optimal model is developed, it undergoes regular training on the greenhouse database, enabling the system to analyze patterns and correlations within the data. Finally, the model makes accurate predictions about future environmental conditions at regular intervals. This workflow will be further detailed in this section.

4.3.1 Data Preprocessing

Data preprocessing is critical in developing AI models, profoundly influencing their effectiveness and performance. This phase, performed on all datasets used in GreenSense AI, encompasses several key operations:

Cleaning: Data cleaning is an important step aimed at identifying and rectifying anomalies, incomplete entries, and irrelevant data within a dataset. Given AI models' sensitivity to noise and inconsistencies, this step is essential. Cleaning involves various techniques such as handling missing data through imputation methods, detecting and correcting outliers, and removing duplicate records to ensure data integrity. Ensuring accuracy and consistency in the dataset enhances the reliability and performance of AI models. Models trained on clean data are more likely to generalize well and make accurate predictions on new, unseen data. In GreenSense AI, errors in the minimum values of multiple features, including wind velocity, were detected in the training dataset, and appropriate handling was taken to rectify these values into a reasonable range.

Feature Engineering: Feature engineering is a pivotal step in data preprocessing where raw data is transformed to create new features or enhance existing ones, thereby improving the predictive power of machine learning algorithms. The goal is to extract meaningful insights relevant to the problem at hand. Techniques include creating new features based on domain knowledge or mathematical transformations, and selecting the most relevant features for modeling.

Effective feature engineering not only boosts model accuracy but also simplifies the learning task by providing more informative input data. By generating features that capture the underlying patterns of the data, this phase empowers machine learning models to make more precise predictions and classifications. In GreenSense AI, sine and cosine functions were used on date and time data to capture important periodic patterns, and the nature of wind velocity was transformed into a format more suitable for model learning, thereby enhancing the model's effectiveness.

Normalization: It is a crucial preprocessing step that ensures all input data features are on a consistent scale, which is essential for many machine learning algorithms. Variations in feature scales can lead to biases in model training, where features with larger ranges dominate the learning process. Common normalization techniques include Min-Max scaling and standardization. By applying normalization, the impact of feature scale differences on model performance is mitigated, leading to more stable and reliable model training. In GreenSense AI, data were normalized using common method: subtracting the mean and dividing by the standard deviation of each feature for each sample. This improves model stability and efficiency, particularly for optimization algorithms like Gradient Descent.

4.3.2 AI Model: Architecture, Training, Validation and Testing Strategies

Following the detailed exploration of data preprocessing, the focus now shifts to model construction. This phase will examine various aspects employed in the development of the model.

Model Design and Selection: For the GreenSense AI approach, a hybrid AI model combining RNN and LSTM was adopted. This architecture harnesses LSTM's capability for long-term memory retention and RNN's proficiency in processing sequential data, aiming to achieve enhanced accuracy and robust forecasting results. To prevent overfitting, the model incorporates multiple dropout layers. Dense layers were employed within this architecture for two primary purposes: enhancing the model's ability to manage complexity and performing linear transformations on the outputs from preceding layers. Additionally, a reshape layer is employed to adjust the output to the desired shape.

“Figure 4.10” presents the model architecture.

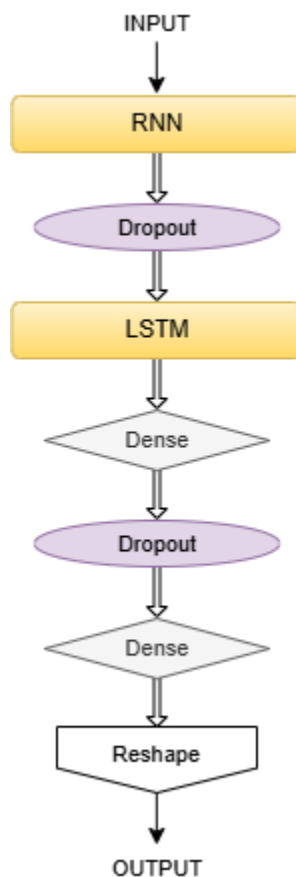


Figure 4.10: GreenSense AI model architecture [62]

This architecture is detailed further in “Table 4.1”

Model layer	Specification
Simple RNN	64 cells with Rectified Linear Unit (ReLU) activation function
Dropout	0.2 dropout rate
LSTM	64 cells with sigmoid activation for gates and tanh activation for candidate state
Dense	8 neurons with ReLU activation function
Dropout	0.2 dropout rate
Dense	Variable number of neurons depending on the desired output shape, using linear activation function
Reshape	Output reshaped to achieve the desired format

Table 4.1: GreenSense AI model architecture details

Data Splitting: During this phase, each dataset undergoes splitting into three subsets: a training set used for model training, an evaluation set for validating training results, and a testing set to assess the obtained model performance. The allocation follows a 70%, 20%, and 10% proportion, respectively.

Model Training and Validation: The model was compiled with various callback functions such as ModelCheckpoint, EarlyStopping, and ReduceLROnPlateau to optimize learning, enhance accuracy, and mitigate overfitting. Mean Squared Error (MSE) served as the loss function, and Mean Absolute Error (MAE) as the evaluation metric.

The training concluded after 13 epochs for the first dataset and 14 epochs for the second dataset, achieving significant progress in both learning and validation processes. For example, “Figure 4.11” displays the training and validation loss curves as well as MAE curves for the second dataset, highlighting the model's learning trajectory with the 9th epoch identified as the optimal epoch.

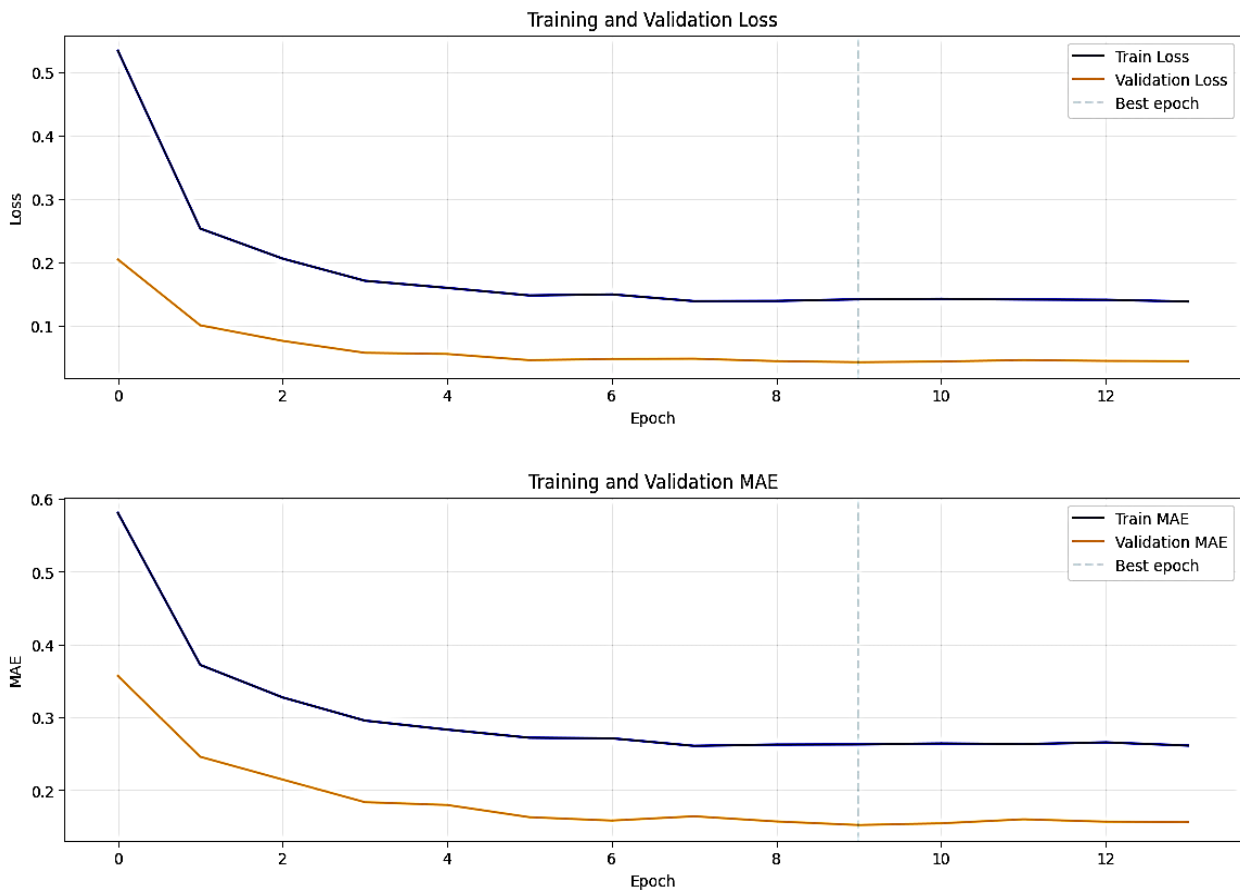


Figure 4.11: Training and validation learning curves

Model Testing: Following training and validation, the best-performing model was loaded and evaluated using the testing set. Predictions generated by the model were compared against actual values. Metrics such as MAE and the R2 coefficient validated the model's efficiency. Furthermore, a comparison plot of predicted versus actual values for one of the forecasted features, as shown in “Figure 4.12”, demonstrates strong match and proves the model's robustness.

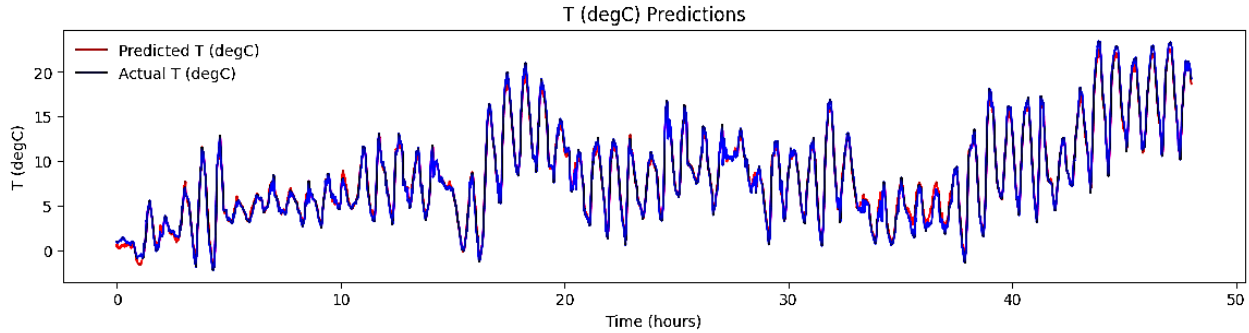


Figure 4.12: Temperature predicted vs actual values

Impact of Callbacks: The effectiveness of callbacks such as ModelCheckpoint, EarlyStopping, and ReduceLROnPlateau in mitigating overfitting and optimizing learning dynamics is noteworthy. ModelCheckpoint ensured that the best-performing model weights were saved based on validation performance, maintaining a record of optimal configurations for subsequent evaluation and deployment. Similarly, EarlyStopping halted training when the validation loss stabilized, avoiding further increases and thereby preventing overfitting while optimizing training efficiency. The ReduceLROnPlateau function dynamically adjusted the learning rate throughout model training based on validation loss trends, which enhanced overall performance. “Figure 4.13” illustrates the impact of these learning rate adjustments and shows how the best epoch was achieved through dynamic learning rate changes, demonstrating their efficacy in optimizing the training process.

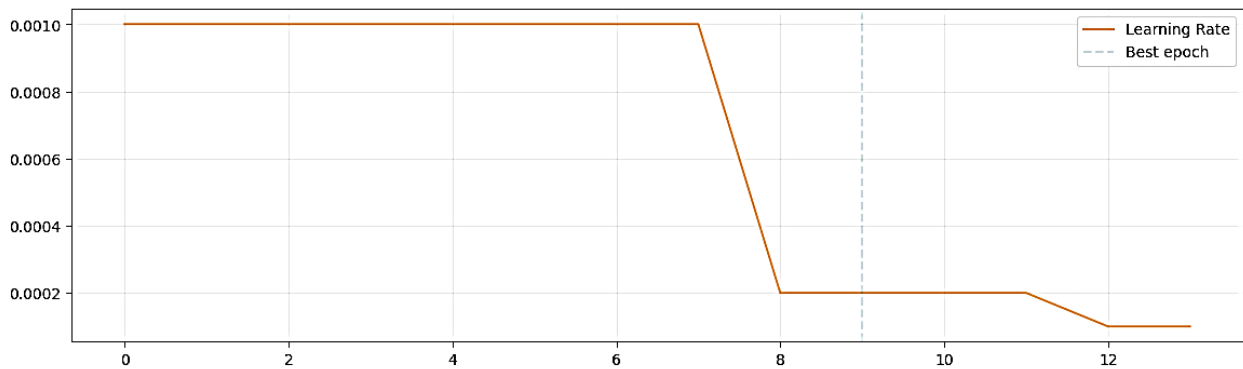


Figure 4.13: Learning rate progress during training

4.3.3 Model Deployment

Following the completion of model development and its validation across multiple datasets, ensuring consistent and reliable performance demonstrating the model's flexibility, the next phase involves deploying the model into its dedicated real-world application. This deployment will occur on a Raspberry Pi device, utilizing two distinct programming codes tailored to handle the datasets generated earlier: the comprehensive and partial databases.

Training and Validation Program: This program replicates the training and validation steps previously outlined. It utilizes the comprehensive database generated by the Raspberry Pi and runs periodically, scheduled for weekly execution. The objective is to continuously enhance the model's performance and adaptability as it learns from accumulating data over time.

Prediction Program: This program is responsible for forecasting future climate parameter values at regular intervals. It operates using the partial database, containing recent collected data, generated by the Raspberry Pi, executing regularly at each time step. These predictions form the basis for recommendations that will be delivered to the farmer.

This deployment strategy ensures that the model remains adaptive and capable of providing timely and accurate forecasts, leveraging real-time data from the Raspberry Pi's databases.

4.4 Insights and Recommendations

In this phase, the focus shifts from model development and deployment to leveraging the model's predictions and analyses for practical applications. The goal is to generate actionable insights and recommendations based on the model's outputs. These insights can empower decision-makers with valuable information to optimize operations, allocate resources efficiently, and enhance overall greenhouse management.

Prediction Evaluation and Recommendations: Once the model generates climate predictions at each time step, these predictions are compared against optimal climate factors specific to the greenhouse's plant species. Based on this comparison, the prediction program generates necessary recommendations alongside the predictions. These recommendations are then published via MQTT for the Node-RED service to receive.

Visualization and Operational Response: Upon subscribing and receiving predictions and recommendations, the Node-RED service visualizes this data on a dashboard, alerting decision-makers to any necessary actions. If actions are required, they can be executed directly from the dashboard interface. Commands are then transmitted from the Node-RED service through the Raspberry Pi and ESP32 to the specific actuators. The actuators promptly adjust conditions within the greenhouse to optimize climate parameters.

As a prototype, the system currently features only two actuators: a fan for temperature and humidity control, and a lamp for light intensity control. “Figure 4.14” depict these components integration within the GreenSense AI system.

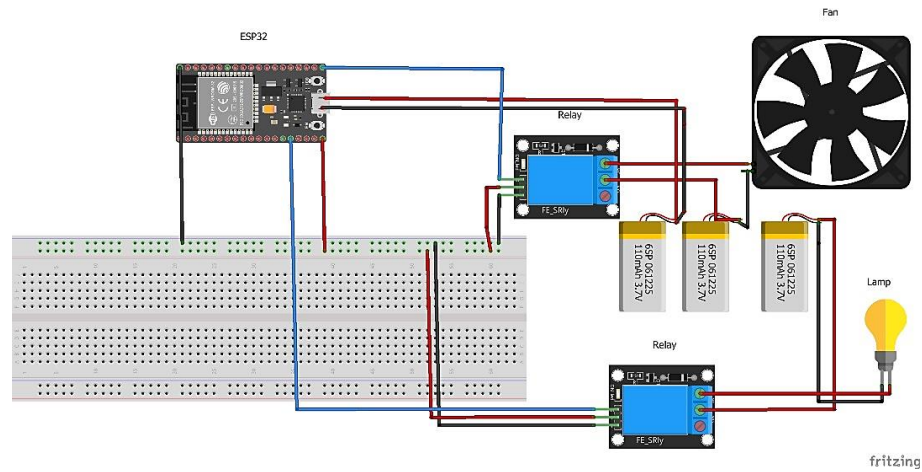


Figure 4.14: Actuators deployment

The full deployment of the interior and exterior nodes is presented in “Figure 4.15” and “Figure 4.16”, respectively

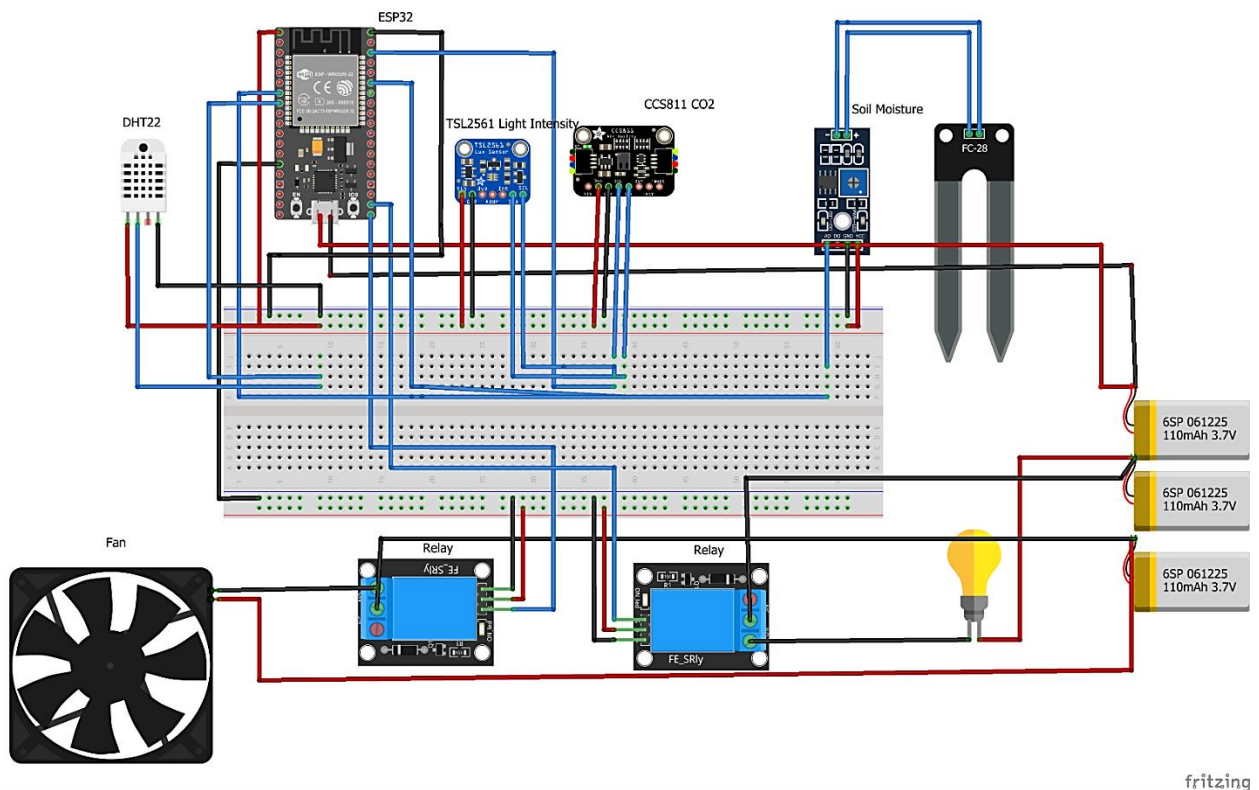


Figure 4.15: Interior node schema

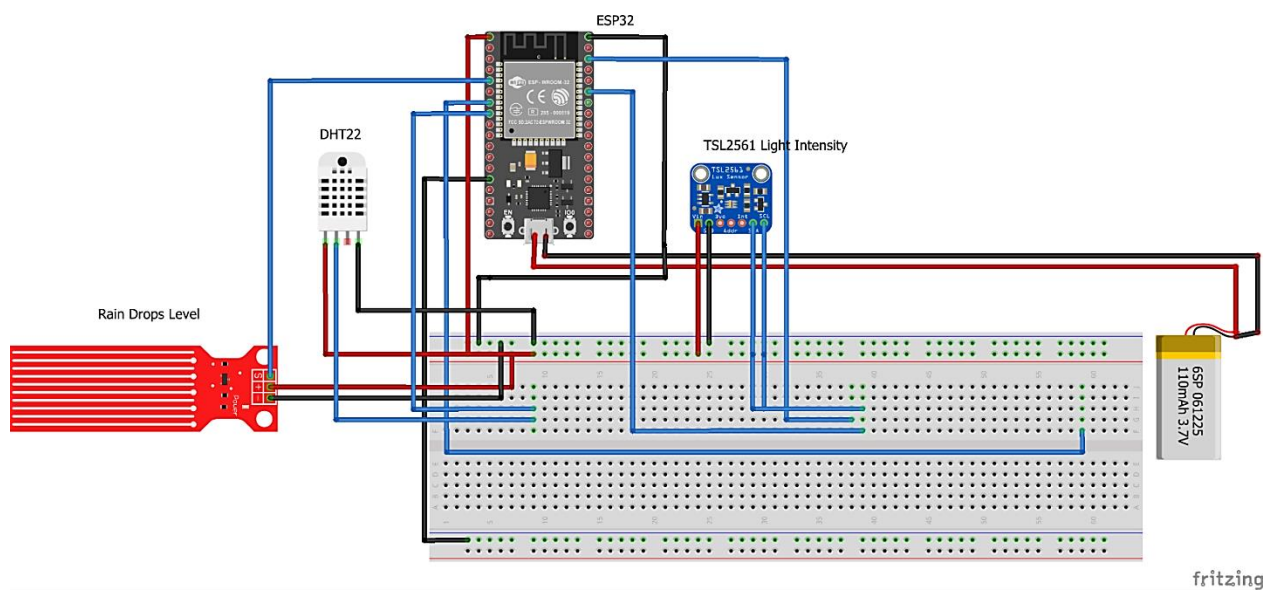


Figure 4.16: Exterior node schema

The flowchart in “Figure 4.17” illustrates the data flow from sensors to the user and back to the actuators.

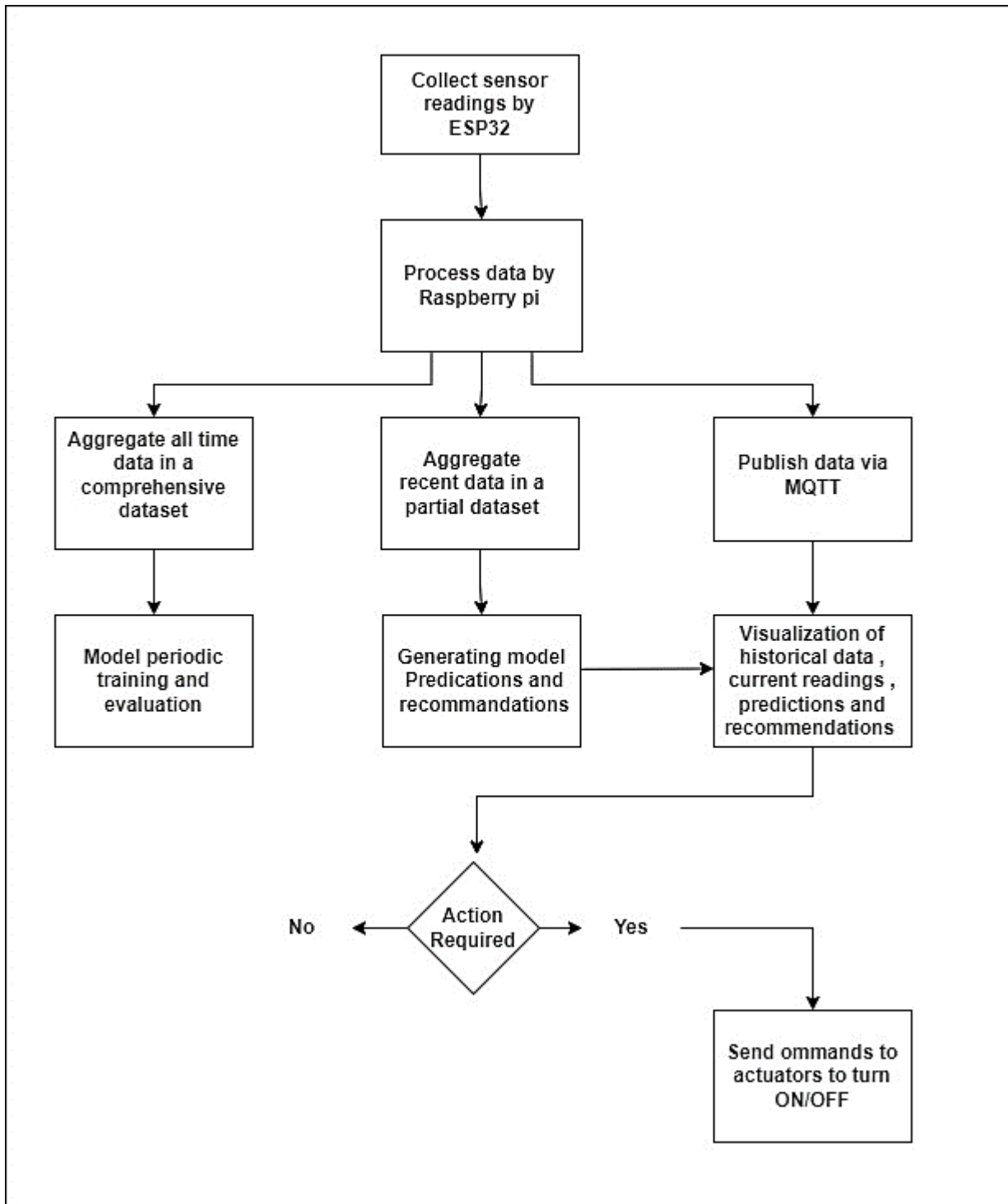


Figure 4.17: Data flow in GreenSense AI

4.5 Results and Discussion

After the implementation phase, it's time to present the results obtained for the project

4.5.1 Prototype

The functional prototype of GreenSense AI proved its practical feasibility in a controlled environment, validating the concept and its applications. “Figure 4.18”, “Figure 4.19”, and “Figure 4.20” show different aspects of GreenSense AI prototype.



Figure 4.19: Inside view of the greenhouse



Figure 4.18: Interior sensing node

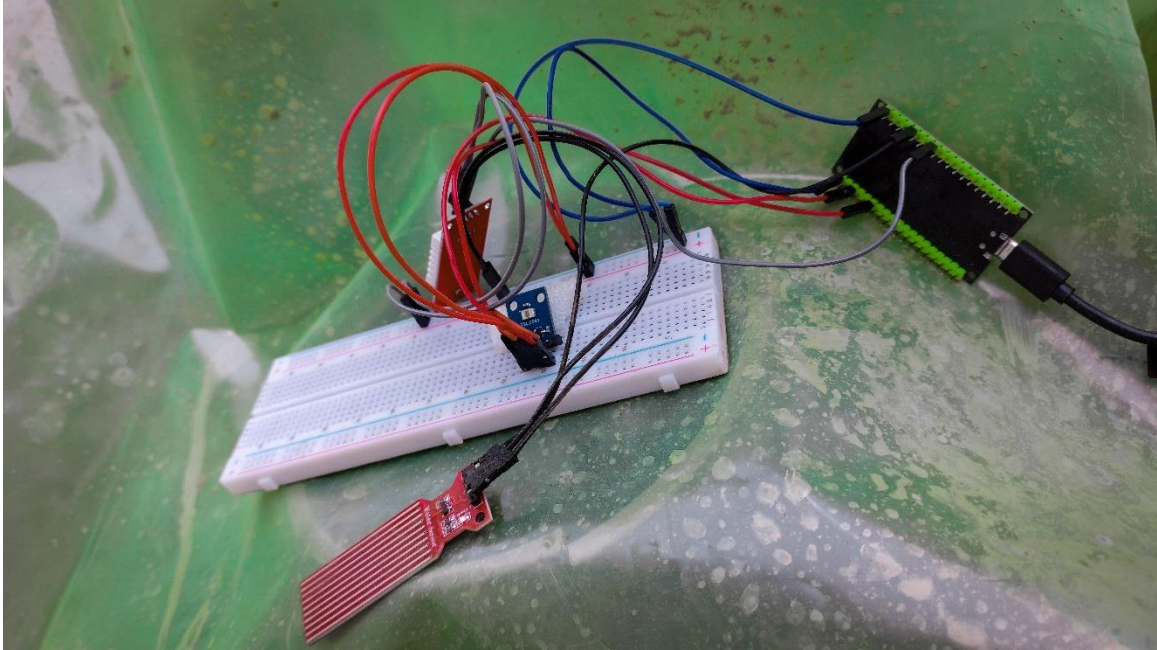


Figure 4.20: Exterior sensing node

4.5.2 Data Acquisition and Management

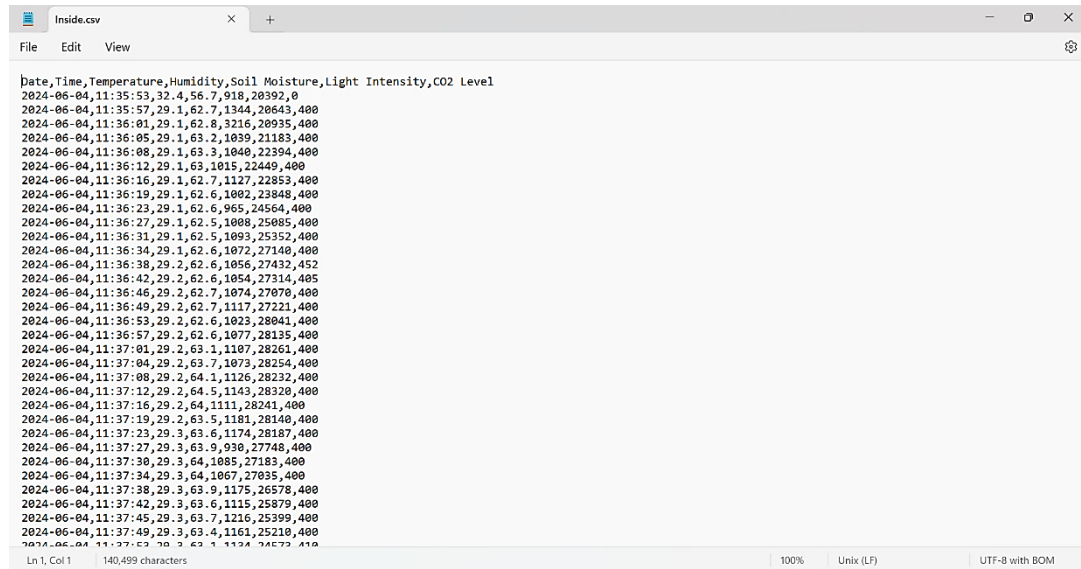
Sensor Measurements: The system provided precise and consistent monitoring of essential parameters such as temperature, humidity, soil moisture, light intensity, CO2 levels, and rain drops. Accurate data collection is crucial for effective greenhouse environment management. “Figure 4.21” provides evidence of these sensor measurements.

```
Received sensor data: {'humidity': 71.40000153, 'raindrops': 4095, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65.83332825, 'co2': 699}
Received sensor data: {'humidity': 71.40000153, 'raindrops': 3971, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65.83332825, 'co2': 907}
Received sensor data: {'humidity': 71.40000153, 'raindrops': 4095, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65.83332825, 'co2': 865}
Received sensor data: {'humidity': 71.40000153, 'raindrops': 4095, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65, 'co2': 885}
Received sensor data: {'humidity': 71.40000153, 'raindrops': 4095, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65, 'co2': 983}
Received sensor data: {'humidity': 71.40000153, 'raindrops': 4095, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65, 'co2': 920}
Received sensor data: {'humidity': 71.5, 'raindrops': 4042, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65, 'co2': 870}
Received sensor data: {'humidity': 71.5, 'raindrops': 4095, 'soilmoisture': 4095, 'temperature': 20.89999962, 'light_intensity': 65, 'co2': 870}
```

Figure 4.21: Sensors readings

Generated datasets: The ESP32 microcontroller efficiently collected and transmitted data to the Raspberry Pi, which then generates the needed datasets.

“Figure 4.22” displays a section of the interior node dataset.



Date	Time	Temperature	Humidity	Soil Moisture	Light Intensity	CO2 Level
2024-06-04	11:35:53	32.4	56.7	918	20392	0
2024-06-04	11:35:57	29.1	62.7	1344	20643	400
2024-06-04	11:36:01	29.1	62.8	3216	20935	400
2024-06-04	11:36:05	29.1	63.2	1039	21183	400
2024-06-04	11:36:08	29.1	63.3	1040	22394	400
2024-06-04	11:36:12	29.1	63.1	1015	22449	400
2024-06-04	11:36:16	29.1	62.7	1127	22853	400
2024-06-04	11:36:19	29.1	62.6	1002	23848	400
2024-06-04	11:36:23	29.1	62.6	965	24564	400
2024-06-04	11:36:27	29.1	62.5	1008	25085	400
2024-06-04	11:36:31	29.1	62.5	1093	25352	400
2024-06-04	11:36:34	29.1	62.6	1072	27140	400
2024-06-04	11:36:38	29.2	62.6	1056	27432	452
2024-06-04	11:36:42	29.2	62.6	1054	27314	405
2024-06-04	11:36:46	29.2	62.7	1074	27070	400
2024-06-04	11:36:49	29.2	62.7	1117	27221	400
2024-06-04	11:36:53	29.2	62.6	1023	28041	400
2024-06-04	11:36:57	29.2	62.6	1077	28135	400
2024-06-04	11:37:01	29.2	63.1	1107	28261	400
2024-06-04	11:37:04	29.2	63.7	1073	28254	400
2024-06-04	11:37:08	29.2	64.1	1126	28232	400
2024-06-04	11:37:12	29.2	64.5	1143	28320	400
2024-06-04	11:37:16	29.2	64.1	1111	28241	400
2024-06-04	11:37:19	29.2	63.5	1181	28140	400
2024-06-04	11:37:23	29.3	63.6	1174	28187	400
2024-06-04	11:37:27	29.3	63.9	930	27748	400
2024-06-04	11:37:30	29.3	64.1	1085	27183	400
2024-06-04	11:37:34	29.3	64.1	1067	27035	400
2024-06-04	11:37:38	29.3	63.9	1175	26578	400
2024-06-04	11:37:42	29.3	63.6	1115	25879	400
2024-06-04	11:37:45	29.3	63.7	1216	25399	400
2024-06-04	11:37:49	29.3	63.4	1161	25210	400

Figure 4.22: Interior node dataset

4.5.3 Climate Prediction

AI Models: The AI models used for climate prediction delivered reliable forecasts, enabling proactive adjustments to the greenhouse climate control systems. This predictive capability was essential for maintaining optimal growing conditions, thereby enhancing crop yield and quality.

4.5.4 Visualization: Insights and Recommendations

The system provides useful insights and recommendations to optimize greenhouse management, helping to monitor and adjust environmental parameters, thereby improving plant growth and yield. “Figure 4.23”, “Figure 4.24”, “Figure 4.25”, “Figure 4.26”, “Figure 4.27”, “Figure 4.28” display the different tabs of the user interface.

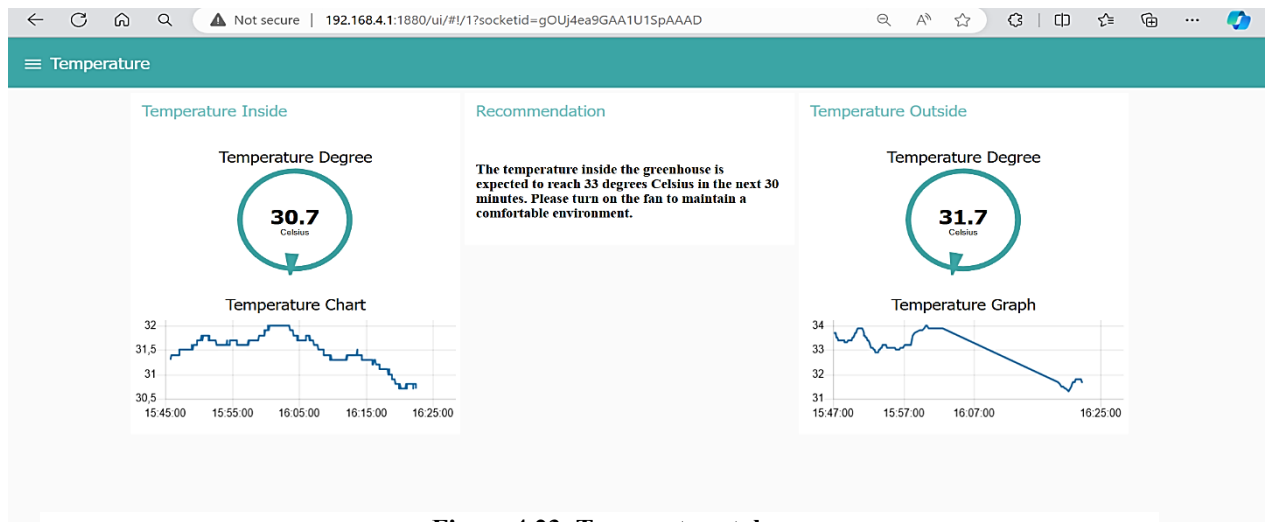


Figure 4.23: Temperature tab

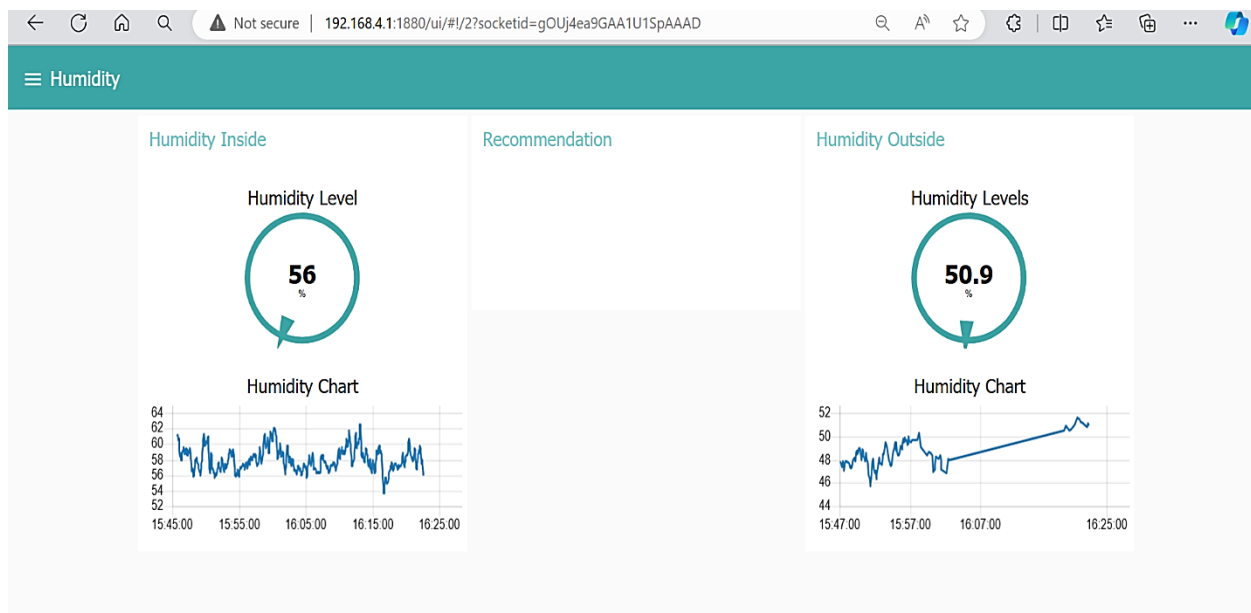


Figure 4.24: Humidity tab

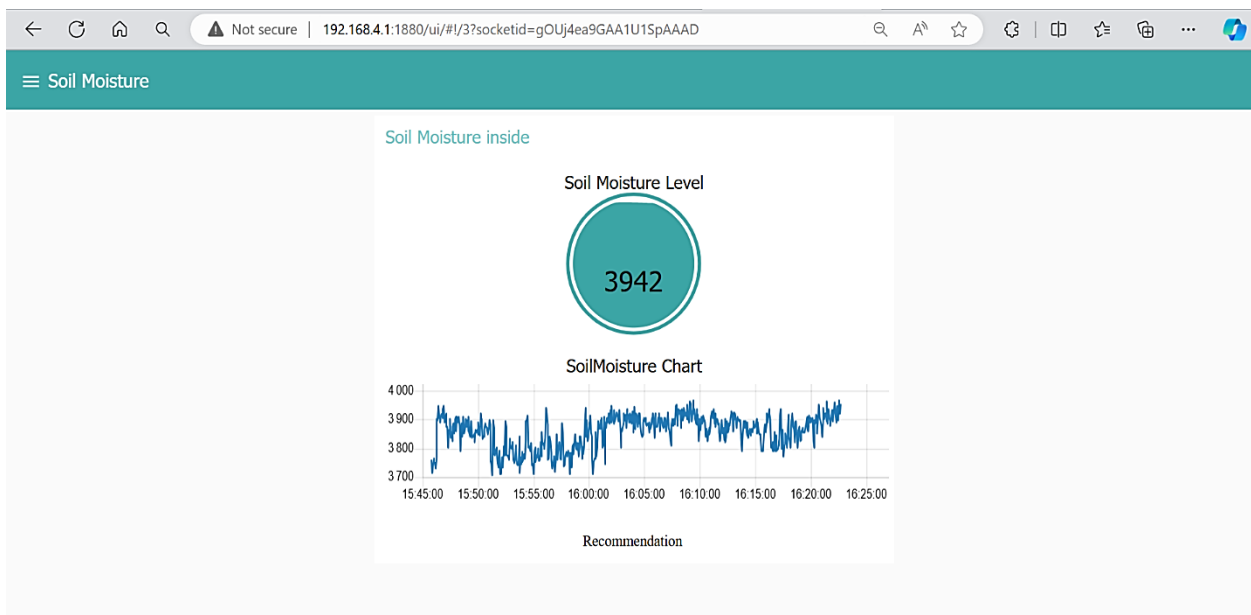


Figure 4.25: Soil moisture tab

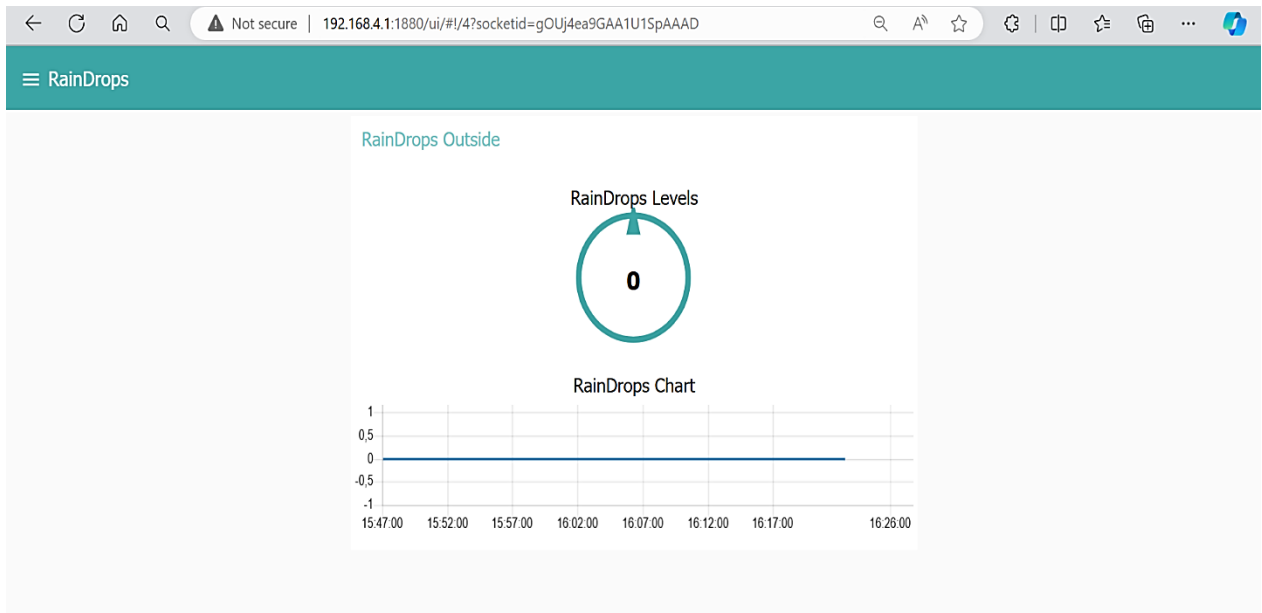


Figure 4.26: RainDrops level tab

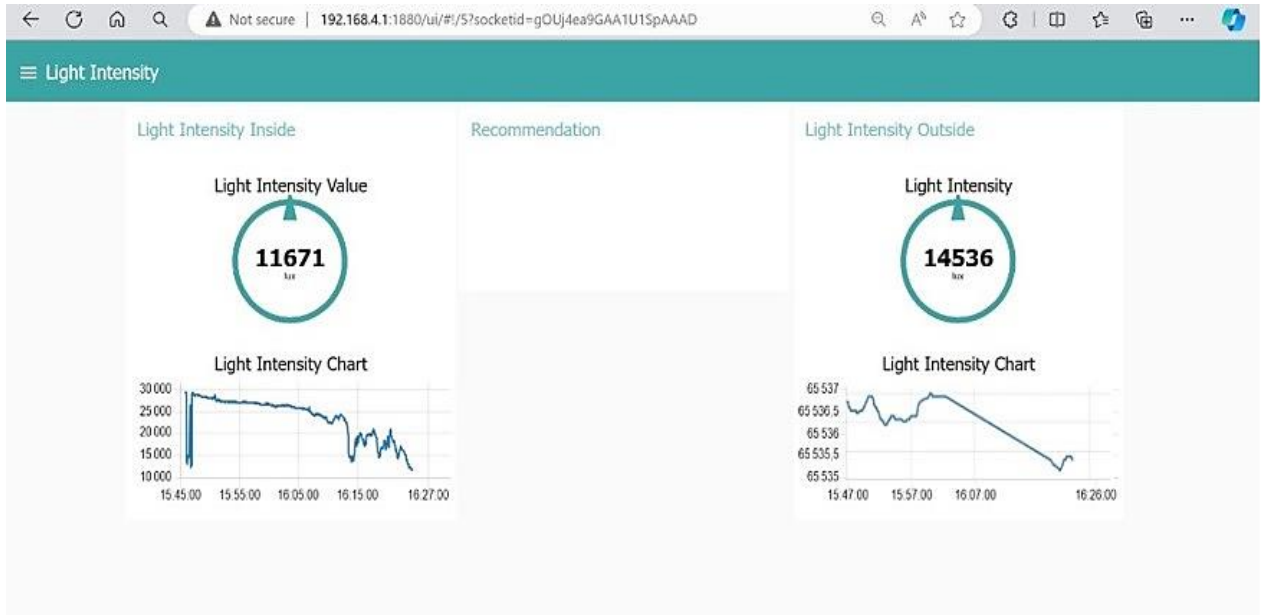


Figure 4.27: Light intensity tab

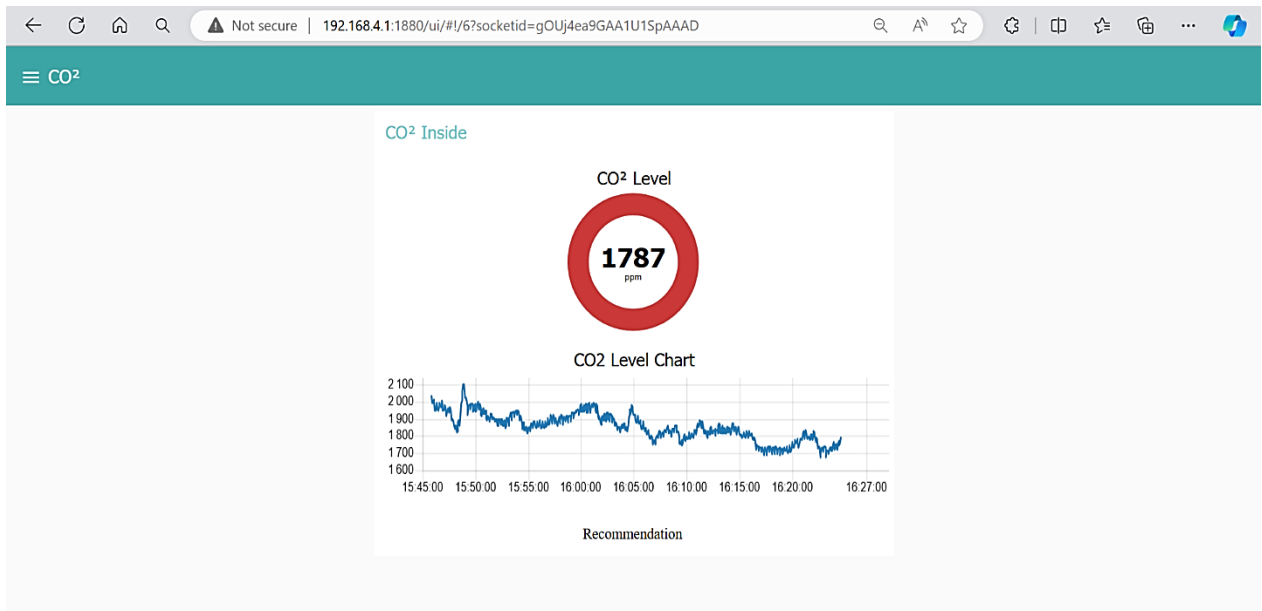


Figure 4.28: CO2 level tab

4.6 Perspective and Future Directions

Scalability of GreenSense AI

The scalability of the GreenSense AI system across multiple greenhouses or agricultural settings envisions deploying a central Raspberry Pi at the farm level, with each greenhouse equipped with its own Raspberry Pi connected to ESP32 and various sensors. Communication within each greenhouse, from ESP32 to Raspberry Pi, will be facilitated via WiFi. To consolidate data across the farm, MQTT will be used for communication between the greenhouse Raspberry Pis and the central Raspberry Pi. Data will then be uploaded continuously to the internet cloud using MQTT from the central Raspberry Pi. This setup will enable access to the system via a website from any location at any time.

“Figure 4.29” simplifies the outlined architecture.

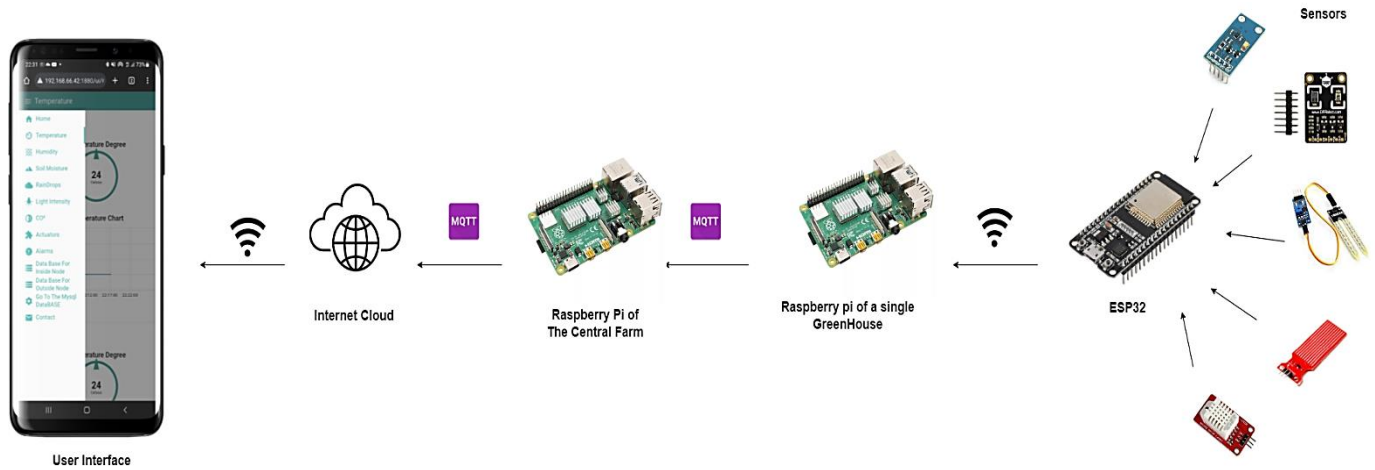


Figure 4.29: Perspective GreenSense AI architecture

Adaptation to Various Agricultural Settings

While the primary focus is on greenhouse climate management, the principles and technologies behind GreenSense AI can be adapted to various agricultural settings. For instance, in open-field agriculture, similar sensor networks and AI-driven analytics could optimize irrigation scheduling and pest management. By expanding into broader agricultural sectors, GreenSense AI has the potential to revolutionize farming practices beyond greenhouse environments.

4.7 Conclusion

The practical implementation of GreenSense AI successfully bridged the gap between theoretical concepts and real-world application, showcasing its potential to revolutionize greenhouse climate management. The results underscored the system's capability to address critical challenges in agriculture, including resource efficiency, crop yield optimization, and sustainable farming practices.

In conclusion, GreenSense AI represents a significant advancement in the integration of IoT and AI technologies within agriculture. The insights and outcomes from this project provide a robust foundation for future developments and broader adoption of similar technologies.

General Conclusion

Feeding a rapidly growing global population sustainably amid climate change and resource constraints presents formidable challenges for agriculture. Conventional farming methods, while effective in the past, are proving insufficient to meet the escalating demands for food production. This project addresses these pressing challenges by harnessing the transformative potential of AI and IoT to develop a cutting-edge smart system specifically designed for greenhouse environments.

The cornerstone of this project lies in IoT technology, which enables the continuous and real-time monitoring of critical environmental parameters within greenhouses. These parameters include temperature, humidity, soil moisture content, light intensity, CO₂ levels, and rainwater levels. By deploying a network of sensors strategically placed throughout the greenhouse, the system captures precise data on these variables. This data is then transmitted to a central hub, typically a Raspberry Pi or similar device, where it is processed and analyzed.

Complementing IoT's data collection capabilities is the advanced analytical prowess of AI. AI algorithms, specifically tailored for agricultural applications, are employed to analyze the vast streams of data generated by IoT sensors. These algorithms can detect patterns, correlations, and anomalies within the data, providing actionable insights into greenhouse conditions and crop health.

The synergy between IoT and AI enables the system to dynamically adjust environmental factors within the greenhouse. For instance, based on real-time data analysis, the system can autonomously regulate temperature and humidity levels to create optimal growing conditions for plants. Similarly, it can adjust irrigation schedules based on soil moisture readings or fine-tune light exposure to maximize photosynthesis efficiency.

One of the most significant contributions of this integrated approach is its ability to empower farmers with actionable insights derived from data. By visualizing trends and predictive analytics through intuitive interfaces (such as Node-RED or custom dashboards), farmers can make informed decisions in real-time. This capability not only enhances operational efficiency but also mitigates risks associated with environmental variability and crop diseases.

The project underscores the paradigm shift towards precision agriculture, where every decision is informed by data-driven intelligence. By optimizing resource use and minimizing environmental impact, precision agriculture promises to revolutionize farming practices. The adoption of AI and IoT in this context represents a crucial step towards achieving these goals, offering scalable solutions that can be customized to fit diverse agricultural settings.

Looking forward, the implications of integrating IoT and AI in agriculture extend beyond individual greenhouse operations. They offer scalable solutions that can be adapted to various farming environments worldwide, from small-scale family farms to large commercial enterprises. By enhancing productivity, resource efficiency, and sustainability, these technologies are poised to play a pivotal role in addressing global food security challenges.

In conclusion, this project exemplifies the transformative potential of IoT and AI in agriculture. By developing a sophisticated yet accessible smart system for greenhouse management, it demonstrates how technology can be harnessed to meet the evolving demands of food production sustainably. As the agricultural sector continues to embrace innovation, the integration of IoT and AI holds promise for shaping a more resilient and productive future for farming.

Appendices

A. Coding

a. Part of the Arduino IDE Code for the Interior Node

```
void loop() {  
  if (!client.connected()) {  
    reconnect();  
  }  
  delay(100);  
  client.loop();  
  WiFiClient client = server.available();  
  delay(1000);  
  value_temperature = dht22.getTemperature();  
  delay(100);  
  value_humidity= dht22.getHumidity();  
  delay(100);  
  value_raindrops = analogRead(34);  
  delay(100);  
  value_soilmoisture = analogRead(32);  
  delay(100);  
  sensors_event_t event;  
  tsl.getEvent(&event);  
  lux = event.light;  
  delay(100);  
  !ccs.readData();  
  value_CO2=ccs.getCO2();  
  StaticJsonDocument<200> doc;  
  doc["humidity"] = value_humidity;  
  doc["raindrops"] = value_raindrops;  
  doc["soilmoisture"] = value_soilmoisture;  
  doc["temperature"] = value_temperature;  
  doc["light_intensity"] = lux;  
  doc["co2"] = value_CO2;  
  String output;  
  serializeJson(doc, output);  
  client.println(output);  
}
```

b. Part of the Arduino IDE Code for the Exterior Node

```
void loop() {
    WiFiClient client = server.available();
    delay(1000);
    value_temperature = dht22.getTemperature();
    delay(100);
    value_humidity= dht22.getHumidity();
    delay(100);
    value_raindrops = analogRead(34);
    delay(100);
    sensors_event_t event;
    tsl.getEvent(&event);
    lux = event.light;
    StaticJsonDocument<200> doc;
    doc["humidity"] = value_humidity;
    doc["raindrops"] = value_raindrops;
    doc["soilmoisture"] = value_soilmoisture;
    doc["temperature"] = value_temperature;
    doc["light_intensity"] = lux;
    doc["co2"] = value_CO2;
    String output;
    serializeJson(doc, output);
    client.println(output);
}
```

c. Part of the Python Code on Raspberry Pi for Interior Node

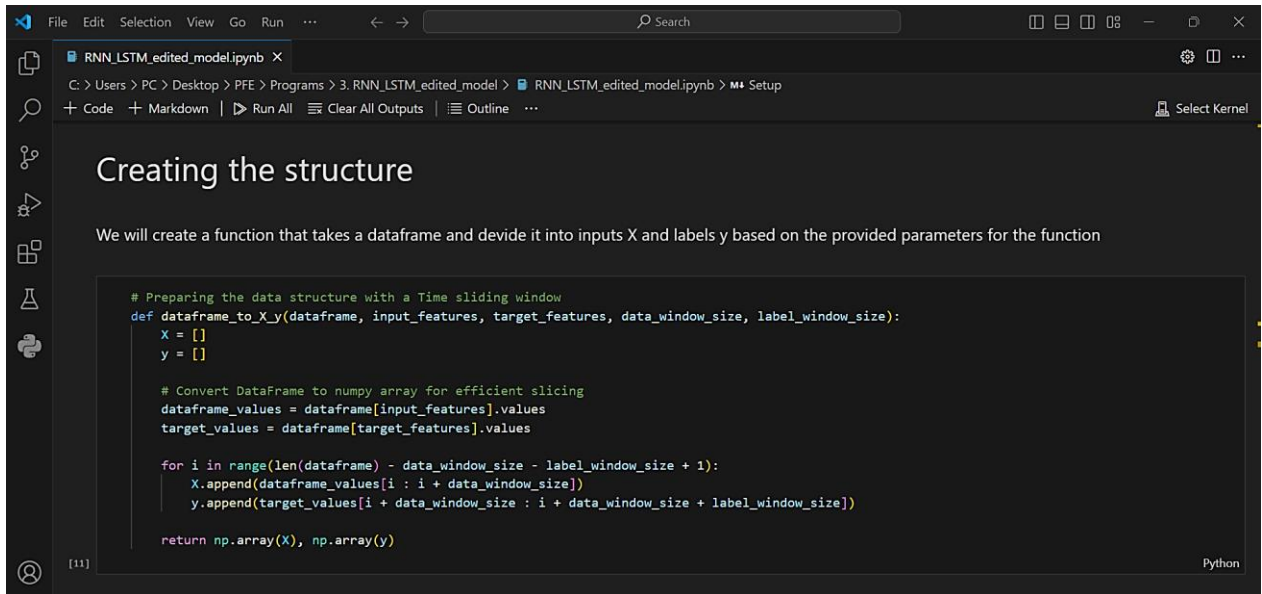
```
def insert_data_to_db(sensor_data):
    cnx = pymysql.connect(**db_config)
    cursor = cnx.cursor()
    query = """INSERT INTO first_node (date, time, temperature, humidity, soilmoisture, raindrops, light_intensity, co2)
    VALUES (CURRENT_DATE, CURRENT_TIME, %s, %s, %s, %s, %s, %s)"""
    data = (sensor_data['temperature'], sensor_data['humidity'], sensor_data['soilmoisture'], sensor_data['raindrops'],
            sensor_data['light_intensity'], sensor_data['co2'])
    cursor.execute(query, data)

    cnx.commit()
    cursor.close()
    cnx.close()

mqtt_client = mqtt.Client()
mqtt_client.connect(mqtt_server, mqtt_port, 60)

def receive_data():
    while True:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((arduino_ip, arduino_port))
        data = s.recv(1024)
        data = data.decode('utf-8')
        try:
            sensor_data = json.loads(data)
            print("Received sensor data:", sensor_data)
            insert_data_to_db(sensor_data)
            for key, value in sensor_data.items():
                mqtt_topic = f"GreenHouse/1st_NODE/{key}"
                mqtt_client.publish(mqtt_topic, str(value))
```

d. Python Code for Creating the Data Structure in AI



File Edit Selection View Go Run ... Search

RNN_LSTM_edited_model.ipynb X

C: > Users > PC > Desktop > PFE > Programs > 3. RNN_LSTM_edited_model > RNN_LSTM_edited_model.ipynb > M Setup

+ Code + Markdown | Run All Clear All Outputs | Outline ...

Select Kernel

Creating the structure

We will create a function that takes a dataframe and divide it into inputs X and labels y based on the provided parameters for the function

```
# Preparing the data structure with a Time sliding window
def dataframe_to_X_y(dataframe, input_features, target_features, data_window_size, label_window_size):
    X = []
    y = []

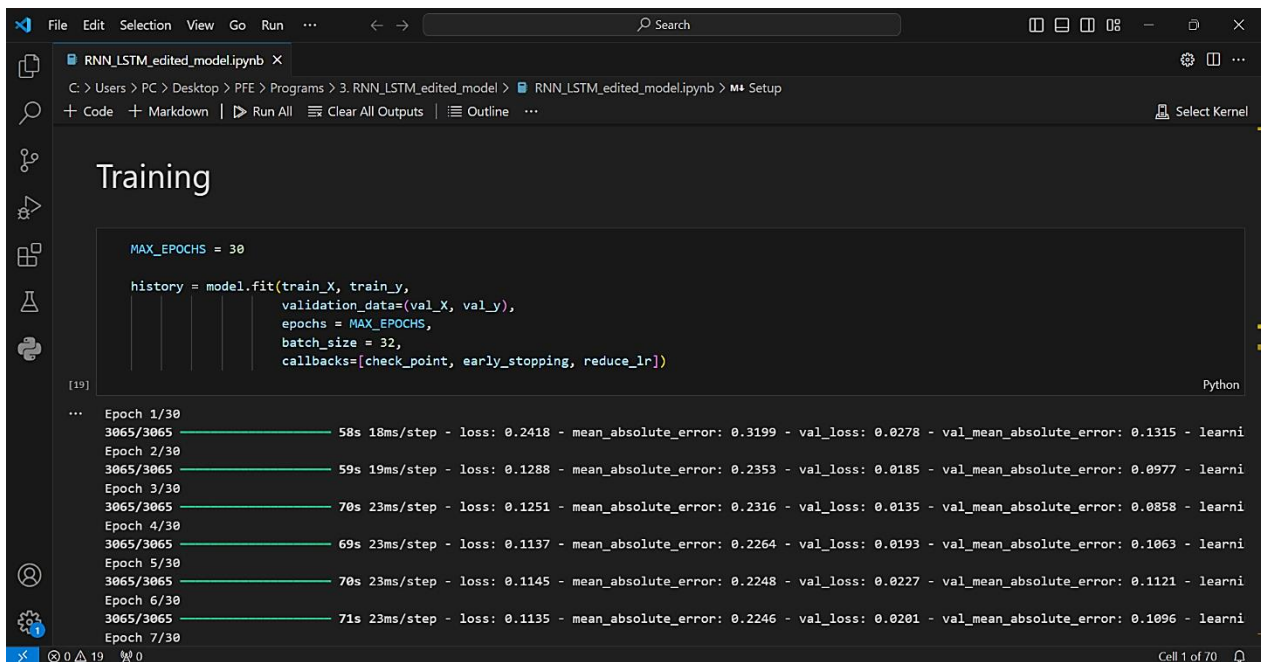
    # Convert DataFrame to numpy array for efficient slicing
    dataframe_values = dataframe[input_features].values
    target_values = dataframe[target_features].values

    for i in range(len(dataframe) - data_window_size - label_window_size + 1):
        X.append(dataframe_values[i : i + data_window_size])
        y.append(target_values[i + data_window_size : i + data_window_size + label_window_size])

    return np.array(X), np.array(y)
```

[11] Python

e. Python Code for the AI Model training launch



File Edit Selection View Go Run ... Search

RNN_LSTM_edited_model.ipynb X

C: > Users > PC > Desktop > PFE > Programs > 3. RNN_LSTM_edited_model > RNN_LSTM_edited_model.ipynb > M Setup

+ Code + Markdown | Run All Clear All Outputs | Outline ...

Select Kernel

Training

```
MAX_EPOCHS = 30

history = model.fit(train_X, train_y,
                    validation_data=(val_X, val_y),
                    epochs = MAX_EPOCHS,
                    batch_size = 32,
                    callbacks=[check_point, early_stopping, reduce_lr])
```

[19] Python

```
... Epoch 1/30
3065/3065 58s 18ms/step - loss: 0.2418 - mean_absolute_error: 0.3199 - val_loss: 0.0278 - val_mean_absolute_error: 0.1315 - learni
Epoch 2/30
3065/3065 59s 19ms/step - loss: 0.1288 - mean_absolute_error: 0.2353 - val_loss: 0.0185 - val_mean_absolute_error: 0.0977 - learni
Epoch 3/30
3065/3065 70s 23ms/step - loss: 0.1251 - mean_absolute_error: 0.2316 - val_loss: 0.0135 - val_mean_absolute_error: 0.0858 - learni
Epoch 4/30
3065/3065 69s 23ms/step - loss: 0.1137 - mean_absolute_error: 0.2264 - val_loss: 0.0193 - val_mean_absolute_error: 0.1063 - learni
Epoch 5/30
3065/3065 70s 23ms/step - loss: 0.1145 - mean_absolute_error: 0.2248 - val_loss: 0.0227 - val_mean_absolute_error: 0.1121 - learni
Epoch 6/30
3065/3065 71s 23ms/step - loss: 0.1135 - mean_absolute_error: 0.2246 - val_loss: 0.0201 - val_mean_absolute_error: 0.1096 - learni
Epoch 7/30
```

Cell 1 of 70

B. Employed Hardware Components Specifications

a. DHT22 Specifications

- Power Supply: 3.3a 6 V
- Max Consumption: 1.5 mA
- Stand by Consumption: 50 μ A
- Measurement Range:
 - Temperature: $-40 + 80$ °C
 - Humidity: 0100%
- Accuracy:
 - Temperature: $\pm 0, 5$ °C
 - Humidity: $\pm 2\%$
 - Dimensions: $25 \times 15 \times 9$ mm

b. SparkFun Specifications

- Power Supply: DC 3.3 - 5 V
- Voltage Signal: 0 ~ 4.2 V
- Current: 35mA
- LED:
 - Power Indicator (Red)
 - Digital Output Indicator (Green)

c. TSL2561 Specifications

- Power Supply: DC 3 – 5 V
- Measurement Range: 1 ~ 65535 lx (lux)
- Current: 0.12 mA
- Communication: I2C bus
- Accuracy: +/-20%

d. CCS811 Specifications

- Power Supply: 1.8V - 3.6 V
- Measurement Range:
 - CO₂: 400 ppm - 8192 ppm
 - TVOC: 0 ppb - 1187 ppb
- Current: Varies depending on operation mode
- Communication: I2C

e. CCS811 Specifications

- Working Voltage: DC 3-5 V
- Working Current: <20mA
- Operating temperature: 10~30 °C
- Sensor Type: Analog
- Detection Area: 40 mm x 16 mm
- Size: 65 mm x 20 mm x 8 mm
- Humidity: 10% -90% non-condensing

f. Sensor connection table

Device	Pin	Interior node	Exterior node
DHT22 Sensor	DATA	33	33
	VCC	3V3	3V3
	GND	GND	GND
Soil Moisture Sensor	VCC	3V3	/
	Digital Out	/	/
	Analog Out	34	/
	GND	GND	/
Fan	/	19	/
TSL2561 Sensor	VCC	3V3	3V3
	SDA	SDA	SDA
	SCL	SCL	SCL
	GND	GND	GND
Lamp	/	4	/
CCS811 Sensor	VCC	3V3	/
	SDA	SDA	/
	SCL	SCL	/
	GND	GND	/
	INT	/	/
	WAKE	GND	GND
	RST	/	/
Rain Drops Sensor	DATA	/	32
	VCC	/	3V3
	GND	/	GND

Bibliography

- [7] E. Baccelli, *Livre blanc internet des objets*, Inria, 2021.
- [9] J. Contreras-Castillo, J. A. Guerrero-Ibanez, P. C. Santana-Mancilla, and L. Anido-Rifon, "SAgric-IoT: An IoT-based platform and deep learning for greenhouse monitoring," *Applied Sciences*, vol. 13, no. 3, p. 1961, 2023.
- [10] S. J. Soheli, N. Jahan, M. B. Hossain, A. Adhikary, A. R. Khan, and M. Wahiduzzaman, "Smart greenhouse monitoring system using Internet of things and artificial intelligence," *Wireless Personal Communications*, vol. 124, no. 4, pp. 3603–3634, 2022.
- [11] C. A. Hernandez-Morales, J. M. Luna-Rivera, and R. Perez-Jimenez, "Design and deployment of a practical IoT-based monitoring system for protected cultivations," *Computer Communications*, vol. 186, pp. 51–64, 2022.
- [12] K. Tatas et al., "Reliable IoT-based monitoring and control of hydroponic systems," *Technologies*, vol. 10, no. 1, p. 26, 2022.
- [13] M. Rabka, D. Mariyanayagam, and P. Shukla, "IoT-based horticulture monitoring system," in *Intelligent Sustainable Systems: Selected Papers of WorldS4 2021*, Singapore: Springer, pp. 765–774, 2022.
- [14] J. Nemčík, E. Mako, and T. Krajcovic, "Smart indoor greenhouse," in *7th Conference on the Engineering of Computer Based Systems*, pp. 1–2, 2021.
- [16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [17] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," in *2017 International Conference on Engineering & MIS (ICEMIS)*, Monastir, Tunisia, 2017, pp. 1–6.
- [19] D. Tilman et al., "Global food demand and the sustainable intensification of agriculture," *Proc. Nat. Acad. Sci.*, vol. 108, no. 50, pp. 20260–20264, 2011.
- [20] H. C. J. Godfray, J. R. Beddington, I. R. Crute, L. Haddad, D. Lawrence, J. F. Muir, et al., "Food security: the challenge of feeding 9 billion people," *Science*, vol. 327, no. 5967, pp. 812–818, 2010.
- [21] V. Sharma, A. K. Tripathi, and H. Mittal, "Technological revolutions in smart farming: Current trends, challenges & future directions," *Comput. Electron. Agric.*, vol. 201, 107217, 2022.
- [24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [25] D. Lunga, "Artificial Intelligence Tools and Platforms for GIS," in *The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2019 Edition)*, J. P. Wilson, Ed., 2019.
- [26] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

- [27] S. Jagtap et al., "The role of artificial intelligence in achieving sustainable development goals," *Nat. Commun.*, vol. 12, no. 1, pp. 1-10, 2021.
- [28] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, pp. 2674, 2018.
- [29] A. A. Mana, A. Allouhi, A. Hamrani, S. Rehman, I. el Jamaoui, and K. Jayachandran, "Sustainable AI-based production agriculture: Exploring AI applications and implications in agricultural practices," *Smart Agric. Technol.*, vol. 7, 100416, 2024.
- [30] C. Maraveas, "Incorporating Artificial Intelligence Technology in Smart Greenhouses: Current State of the Art," *Appl. Sci.*, vol. 13, p. 14, 2023.
- [31] M. El Jarroudi, L. Kouadio, P. Delfosse, et al., "Leveraging edge artificial intelligence for sustainable agriculture," *Nat. Sustain.*, 2024.
- [34] C.-K. Chou, C.-C. Lo, S.-H. Huang, M.-F. Horng, and Y.-H. Kuo, "Greenhouse Climate Control Based on Time Series Analysis," in *2018 IEEE World Congress on Services (SERVICES)*, San Francisco, CA, USA, pp. 61-62, 2018.
- [35] J. D. Hamilton, *Time Series Analysis*, Jan. 31, 1994.
- [36] Y. Chen and J. Li, "Recurrent Neural Networks algorithms and applications," in *2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, Zhuhai, China, pp. 38-43, 2021.
- [37] K. Smagulova and A. P. James, "Overview of Long Short-Term Memory Neural Networks," in *Deep Learning Classifiers with Memristive Networks*, A. James, Ed., *Modeling and Optimization in Science and Technologies*, vol. 14. Cham, Switzerland: Springer, pp. 307-329, 2020.
- [40] Z. Liu, J. Mei, D. Wang, Y. Guo, and L. Wu, "A Novel Damage Identification Method for Steel Catenary Risers Based on a Novel CNN-GRU Model Optimized by PSO," *J. Mar. Sci. Eng.*, vol. 11, no. 1, p. 200, Jan. 12, 2023.
- [45] S. Karle, V. Bansode, P. Tambe, and R. Bhambare, "IoT Based Greenhouse Monitoring System Using Raspberry Pi," *International Journal of Scientific Research in Science and Technology*, pp. 360-367, 2021.
- [46] V. M. Quan, G. Sen Gupta, and S. Mukhopadhyay, "Review of sensors for greenhouse climate monitoring," in *2011 IEEE Sensors Applications Symposium*, San Antonio, TX, USA, pp. 112-118, 2011.
- [52] D. -I. Săcăleanu, Ș. -G. Roșu, M. -G. Matache, I. -P. Manciu, and L. -A. Perișoară, "Wireless actuator node for greenhouse microclimate control," in *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Ploiesti, Romania, pp. 1-4, 2022.
- [57] A. Joy, S. Thangavelu, and A. Jyotishi, "Performance of GitHub Open-Source Software Project: An Empirical Analysis," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 794-799, 2018.
- [62] S. I. Zeghdar and M. C. Souiki, "Greenhouse Climate Forecasting: an RNN-LSTM approach," *National Higher School of Advanced Technologies*, Algiers, Algeria, 2024.

Webography

- [1] "World Population Projected to Reach 9.8 Billion in 2050 and 11.2 Billion in 2100," United Nations Department of Economic and Social Affairs, 2017. [Online]. Available: <https://www.un.org/tr/desa/world-population-projected-reach-98-billion-2050-and-112-billion-2100>. [Accessed 10, 06, 2024].
- [2] "The State of the World's Land and Water Resources for Food and Agriculture: Managing Systems at Risk," 2011. [Online]. Available: <https://www.fao.org/4/i1688e/i1688e.pdf>. [Accessed 10 06 2024].
- [3] "How to Feed the World in 2050 (High-Level Expert Forum)," Food and Agriculture Organization of the United Nations, 2009.
- [4] "Sustainable Management of Water Resources in Agriculture," Organisation for Economic Co-operation and Development, 2010.
- [5] "L'INTERNET DES OBJETS (IOT) POUR UNE AGRICULTURE PLUS INTELLIGENTE," [Online]. Available: <https://www.mutualia.fr/agriculteur/infos/economie-et-societe/news/internet-des-objets-iot-pour-une-agriculture-plus-intelligente>. [Accessed 05 06 2024].
- [6] "Internet of Things: The Protocols Landscape," Electronics For You, 16 04 2021. [Online]. Available: <https://www.electronicsforu.com/technology-trends/tech-focus/internet-things-protocols-landscape>. [Accessed 05, 06, 2024].
- [8] Jaxel, "3 IoT Applications That Will Save Your Business Money," Jaxel, Oct. 6, 2021. [Online]. Available: <https://jaxel.com/3-iot-applications-that-will-save-your-business-money/>
- [15] Steve's Internet Guide, "MQTT Protocol - Messages Overview," Updated: Apr. 11, 2023. [Online]. Available: <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>
- [18] EMQX Blog, "Authorization in MQTT: Using ACLs to Control Access to MQTT Messaging," 2023. [Online]. Available: <https://github.com/emqx/blog/blob/main/en/202308/authorization-in-mqtt-using-acls-to-control-access-to-mqtt-messaging.md>
- [22] McKinsey & Company, "What is AI (artificial intelligence)?," April 2024. [Online]. Available: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-ai>
- [23] TechTarget, "What is artificial intelligence (AI)? Everything you need to know," April 2024. [Online]. Available: <https://www.techtarget.com/searchenterpriset/definition/AI-Artificial-Intelligence>
- [32] Investopedia, "What Is a Time Series and How Is It Used to Analyze Data?," Retrieved from: <https://www.investopedia.com/terms/t/timeseries.asp>
- [33] O. Gerasymov, "Machine Learning For Time Series Forecasting," CodeIT, Jan. 4, 2023. [Online]. Available: <https://codeit.us/blog/machine-learning-time-series-forecasting>
- [38] vrogue.co, "Structure of a basic LSTM block - Download scientific diagram," [Online]. Available: <https://www.vrogue.co/post/structure-of-a-basic-lstm-block-download-scientific-diagram>. [Accessed: Jun. 10, 2024].
- [39] S. Saxena, "Introduction to Gated Recurrent Unit (GRU)," Analytics Vidhya, Jun. 2, 2024. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/>. [Accessed: Jun. 10, 2024].

- [41] D. Reynolds, "ESP32 vs. STM32: How to Choose the Right Microcontrollers," xecor, Jan. 11, 2024. [Online]. Available: <https://www.xecor.com/blog/esp32-vs-stm32>. [Accessed: Jun. 10, 2024].
- [42] D. Reynolds, "ESP32 vs. ESP8266: How to Choose the Right Microcontrollers," xecor, Jan. 29, 2024. [Online]. Available: <https://www.xecor.com/blog/esp32-vs-esp8266>. [Accessed: Jun. 14, 2024].
- [43] Espressif, "ESP32 for IoT: A Complete Guide," nabto, 2024. [Online]. Available: <https://www.nabto.com/guide-to-iot-esp-32/>. [Accessed: Jun. 14, 2024].
- [44] doritique, "Articles," doritique, 2024. [Online]. Available: https://www.doritique.fr/Articles/View_Article.php?num_article=95. [Accessed: Jun. 16, 2024].
- [47] Aosong Electronics Co., Ltd, "Capteurs de température et d'humidité DHT22," [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/download/1132459/ETC2/DHT22.html>. [Accessed: Jun. 16, 2024].
- [48] Components 101, "Soil Moisture Sensor Module," components 101, Apr. 16, 2020. [Online]. Available: <https://components101.com/modules/soil-moisture-sensor-module>. [Accessed: Jun. 16, 2024].
- [49] AMSCO (ams AG), "TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER TAOS059N," Mar. 2009. [Online]. Available: <https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf>. [Accessed: Jun. 16, 2024].
- [50] CCS811 Ultra-Low Power Digital Gas Sensor for Monitoring Indoor Air Quality ams Datasheet [v1-05], May 1, 2018. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1047395/AMSCO/CCS811.html>. [Accessed: Jun. 16, 2024].
- [51] BIOMAKER CHALLENGE, "Water Level Sensor," Biomaker.org, Dec. 17, 2021. [Online]. Available: <https://www.biomaker.org/block-catalogue/2021/12/17/water-level-sensor-tzt-water-level-sensor>. [Accessed: Jun. 16, 2024].
- [53] RealVNC, "All you need to know about VNC remote access technology," discover.realvnc, 2022. [Online]. Available: https://discover.realvnc.com/what-is-vnc-remote-access-technology?lai_sr=30-34&lai_sl=m. [Accessed: Jun. 16, 2024].
- [54] Eclipse Mosquitto™, "Home," mosquitto, 2024. [Online]. Available: <https://mosquitto.org>. [Accessed: Jun. 16, 2024].
- [55] Arduino, "About," arduino, Sep. 15, 2021. [Online]. Available: <https://www.arduino.cc/en/about>. [Accessed: Jun. 16, 2024].
- [56] Microsoft, "Learn," Visual Studio Code, 2024. [Online]. Available: <https://code.visualstudio.com/learn>. [Accessed: Jun. 17, 2024].
- [58] Oracle, "MySQL," 2024. [Online]. Available: <https://dev.mysql.com/doc/refman/8.4/en/>. [Accessed: Jun. 17, 2024].
- [59] Node-RED Team, "About," Node-RED. [Online]. Available: <https://nodered.org/about/>. [Accessed: Jun. 18, 2024].
- [60] Python Software Foundation, "About," Python. [Online]. Available: <https://www.python.org/about/>. [Accessed: Jun. 18, 2024].

- [61] T. Kluyver et al., "Jupyter Notebooks – a publishing format for reproducible computational workflows," in International Conference on Electronic Publishing, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:36928206>