EOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION

AND SCIENTIFIC RESEARCH

SCIENTIFIC

YAHIA FARES UNIVERSITY OF MEDEA



Faculty of Technology Department of Electrical Engineering

Master's degree dissertation

Speciality: Systems of Telecommunications.

Home daily living activity monitoring for elderly

Submitted by:

HERICHE Houda

Supervised by:

- Dr. TOUBAL Abdelmoughni
- Dr. BENGHERBIA Billel

ACADEMIC YEAR

2022/2022

Acknowledgments

Before we begin, praise and gratitude to Allah the Almighty, for having us given the will, the health, the Patience and the courage to achieve our goal.

I would like to express my sincere gratitude to my supervisors, Dr. TOUBAL Abdelmoughni and Dr. BENGHERBIA Billel, for their invaluable guidance and support throughout my project. In particular, I would like to extend my heartfelt appreciation to Dr. TOUBAL Abdelmoughni, who, despite his demanding responsibilities, dedicated his time and expertise to meticulously oversee and refine my work. I deeply appreciated his unwavering presence throughout the entire year. His exceptional patience, insightful advice, technical and material assistance, as well as constant encouragement, deserve our utmost appreciation."

Furthermore, we extend our thanks to PHD students Mr. TOBBAL Abdelhafid and Mr. BERKANI Mohamed Rafik Aymene for their help to the success of this project.

"We also wish to express our gratitude to the members of the jury, Mr. ESSAID Billel and Mr. LACHENANI Sidahmed, for their presence and evaluation of our work. Their expertise and feedback have been invaluable in shaping our future projects."

And thanks to all the participants of this research, whose cooperation and willingness to share their experiences and perspectives made this study possible. Their contributions have added depth and richness to the findings of this thesis.

I would like to express my sincere gratitude to my family and friends, for their unwavering support, encouragement, and understanding throughout this demanding journey. Their belief in me has been a constant source of motivation.

Finally, I would like to extend my deepest appreciation to all the individuals who have directly or indirectly contributed to the completion of this thesis. Your support and encouragement have been invaluable, and I am grateful for your presence in my academic and personal life.

Dedication

I dedicate this master's thesis to:

My father and mother, for their unconditional love, unwavering support, and constant encouragement. Your belief in me has been the driving force behind my pursuit of knowledge and personal growth. This achievement would not have been possible without your sacrifices and guidance.

My sisters Marwa, Yousra, Sarah, Yasmine, my brother Zakaria and each one in the family HERICHE and MENASRI who have always been my pillars of strength and unwavering support throughout my academic journey. Thank you for believing in me.

my fiancé MOOGAFI Hamza, who has been my cheerleader, and my source of unwavering support throughout this entire master's thesis journey. Your belief in my abilities, your constant encouragement, and your understanding during the stressful times have been invaluable to me.

My best friend’s Afrah, Roufaida, Samer, Rahma,Asmaa, Ikbal, Hanane and Imane, for their companionship, understanding, and motivation during the challenging times of this academic endeavor. Your unwavering support and shared experiences have made this journey more enjoyable and memorable.

My Mentor Dr. TOUBAL Abdelmoughni, for his mentorship, guidance, and invaluable insights that have shaped my academic and professional development. Their wisdom and encouragement have been pivotal in my growth as a researcher and scholar.

The participants of my study, who generously shared their time, knowledge, and experiences. Your willingness to contribute to this research has made a significant impact on the outcomes of this thesis, and I am deeply grateful for your invaluable insights.

Lastly, I would like to dedicate this thesis to all those who believe in the power of education, research, and intellectual pursuits.

الملخص:

يهدف هذا المشروع إلى تطوير نظام شامل لمراقبة أنشطة الحياة اليومية لكبار السن باستخدام تقنية الذكاء الاصطناعي (AI). يستخدم النظام أجهزة إنترنت الأشياء وأجهزة الاستشعار وتحليلات البيانات لتوفير رؤى قيمة حول رفاهية وسلامة المسنين داخل منازلهم لتمكين المراقبة الفعالة في الوقت الفعلي ، يستخدم المشروع البطاقة الإلكترونية ESP32. يتم استخدام مستشعرين MPU6050 لتوفير استشعار الحركة وتتبع الاتجاه وقياس التسارع والسرعة الزاوية . استخدمنا خوارزميات التعلم الآلي وتحديداً نموذج الشبكة العصبية التلافيفية 1D (CNN) ، من اجل معالجة البيانات وتحليلها و تصنيف الأنشطة اليومية للشخص الذي تتم مراقبة نشاطه.

الكلمات المفتاحية: الشبكة العصبية التلافيفيةCNN , IOT, AI, MPU6050, ESP32, انسان مسن,المتابعة الزمنية.

The abstract:

This project aims to develop a comprehensive system for monitoring the daily activities of elderly individuals using Artificial Intelligence (AI) technology. The system utilizes Internet of Things (IoT) devices, sensors, and data analytics to provide valuable insights on the well-being and safety of seniors within their homes, enabling effective real-time monitoring. The project utilizes the ESP32 microcontroller. MPU6050 sensors are employed to provide motion sensing, orientation, acceleration measurement, and angular velocity. We utilized machine learning algorithms, specifically the 1D Convolutional Neural Network (CNN) model, to process, analyze, and classify the monitored individual's daily activities.

Key words: IOT, AI, MPU6050, ESP32, elderly, CNN, monitoring,

Le résumé :

Ce projet vise à développer un système complet pour surveiller les activités quotidiennes des personnes âgées en utilisant la technologie de l'intelligence artificielle (IA). Le système utilise des appareils IoT, des capteurs et des analyses de données pour fournir des informations précieuses sur le bien-être et la sécurité des personnes âgées dans leur domicile, permettant une surveillance en temps réel efficace. Le projet utilise la carte électronique ESP32. Les capteurs MPU6050 sont utilisés pour détecter les mouvements, suivre l'orientation et mesurer l'accélération et la vitesse angulaire. Nous avons utilisé des algorithmes d'apprentissage automatique, en particulier le modèle de réseau de neurones convolutifs 1D (CNN), pour traiter, analyser et classifier les activités quotidiennes de la personne surveillée.

Mots clés : IOT, AI, MPU6050, ESP32, personnes âgées, CNN, surveillance

Table of content

List of Figures.....10
List of Tables.....13
List of Abbreviations.....14

General introduction

chapter I: An overview of Artificial Intelligence

I.1 INTRODUCTION.....3
I.2.Definition of artificial intelligence.....3
I.3.History of AI.....3
I.4.The different approaches of AI.....5
a) Acting humanly (The Turing Test approach).....5
b) Thinking humanly (The Cognitive approach).....5
c) Thinking rationally (The laws of thought approach).....5
d) Acting rationally (The rational agent approach).....5
This approach gives us four possible goals in artificial intelligence:.....6
I.5.Applications of AI.....6
I.5.1 AI in healthcare.....7
I.5.2 AI in Astronomy.....7
I.5.3 AI in Finance.....7
I.5.4 AI in Data Security.....7
I.5.5 AI in social media.....7
I.5.6 AI in Travel & Transport:.....7
I.5.7 AI in Automotive Industry.....8
I.5.8 AI in Robotics.....8
I.5.9 AI in Entertainment.....8
I.5.10 AI in Agriculture.....8
I.5.11 AI in E-commerce.....8
I.5.12 AI in education.....8
I.6.Machine Learning.....9
I.6.1. Definition machine Learning.....9
I.6.2. The different types of machine learning.....9

a) Supervised learning 9
b) Unsupervised learning 10
c) Reinforcement learning 10
I.7.Deep Learning 11
I.7.1. Artificial Neural Networks (ANNs) 12
a) Definition 12
b) Architecture of Artificial Neural Network 12
c) The functioning of artificial neural networks 13
I.7.2 Convolution neural networks 14
a) convolutional layers 15
b) pooling layers 16
c) Fully-connected layers 18
d) Output Layer: 18
I.8. Conclusion 19

chapter II: wireless Sensor Network and IOT.

II.1 Introduction 20
II.2 Wireless Sensor Networks (WSN) 20
II.2.1 Definition 20
II.2.2 WSN Architecture 21
II.2.3 The different components of WSN 21
a) The sensor nodes 21
b) Gateway 21
c) Software WSNs 21
II.3 Internet of Things (IOT) 22
II.3.1 definition 22
II.3.2 IoT Architecture 22
a) The main layers in IoT architecture: 22
b) The different components in IOT 23
II.4 The different applications of wireless sensor networks and IoT 23
II.4.1 Healthcare: 23
a) Remote patient monitoring: 24
b) Ambient assisted living: 24
c) Medication management 24
II.4.2 Agriculture 24

a) Soil Monitoring 25
b) Crop Monitoring.....25
c) Management.....25
d) Livestock Monitoring 25
e) Precision Agriculture.....25
II.4.3 Transportation.....25
a) Smart Traffic Management 25
b) intelligent transportation systems (ITS): 26
c) smart parking: 26
d) Condition monitoring: 26
e) Fleet Management: 26
II.4.4 smart cities: 26
II.5 The different communication protocols used in WSN and IOT 27
II.5.1 ZigBee.....27
II.5.2 Bluetooth.....28
II.5.3 Wi-Fi 29
II.5.4 LoRaWAN.....30
II.5.5 MQTT 30
II.6 Energy Efficiency 31
II.6.1. Techniques used to extend the battery life of sensors nodes:.....31
- Dynamic Voltage Frequency Scaling (DVFS) 32
- duty cycling:.....32
- Sleep scheduling.....32
- Energy Harvesting.....32
- Cognitive Radio (CR).....32
- Data redaction 32
- Data compression 33
- Data aggregation 33
- Data prediction.....33
- Low-Power Components 33
- using energy-efficient communication protocols, ...33

II.7 Security and Privacy 33
II.7.3 Data Integrity.....34
II.7.4 Availability 34

II.8 Conclusion 35

chapter III: Design and implementation of the system, tests and results.

III.1 Introduction 37
III.2 Project conception 37
III.3 Description of the components used. 38
III.3 .1 Esp32 39
III.3.2 MPU 6050: 41
a) The Gyroscope 41
b) The accelerometer: 42
c) Block diagram of the MPU6050 circuit 42
III.4 System architecture 43
III.4.2 The communication protocol between ESP32 and MPU6050 44
III.5 The software used 45
III.5 .1 The Arduino IDE 45
a) The utilisation and configuration of Arduino ide 45
b) The installation of ESP32 carte in Arduino IDE 46
c) Installation of the library of MPU6050 47
III.5 .2 Python 48
III.5 .3 Libraries used 49
a) 1.TensorFlow 49
b) Keras 49
c) sklearn 50
d) NumPy 50
e) Matplotlib 50
III.6 The dataset 51
III.6.1 Dataset preparation 51
III.6.2 data description 51
III.6.3. CSVs file Description 52
III.7 The CNN model 55
III.7.1 Import packages. 55
III.7.2 Pre-processing of input data from CSV files in each directory 56
1) Decomposition of data 57
III.7.3 Architecture used 57
III.7.4 Save the best model 61

III.7.5 The obtained results.....62
a) Accuracy and losses graph.....62
b) Confusion Matrix 63
III.8 Implementation.....64
III.8.1 Model preparation 64
III.8.2 Implementation result.....64
III.9 conclusion.....67
General conclusion 69

List of Figures

Figure1.1: the main application of AI 06

Figure I.2: Main types of machine learning 11

FigureI.3: Deep learning is subset of machine learning.....11

Figure I.4: The different layers of artificial neural network.....12

Figure I.5: Artificial neural networks Works 14

Figure I.6: Convolution neural network 15

Figure I.7: One step in the Convolution Process. 15

Figure I.8: Convolutional layers works With and without zero padding.....16

Figure I.9: The pooling layers.....17

Figure I.10: The types of pooling layers 17

Figure I.11 Fully-connected layers 18

Figure II. 1: Architecture of a wireless sensor node. 20

Figure II.2: WSN architecture 21

Figure II.3: WSN in healthcare application.....24

Figure II.4: Soil quality Monitoring 25

Figure II.5: WSNs and IoT in smart Traffic Management.....26

Figure II.6: WSN and IOT in smart cities 27

FigureII.7: Wireless sensor and actor networks based on ZigBee protocol.....28

Figure II.8: Bluetooth Low Energy Applications in IOT.....29

Figure II.9: Wi-Fi application in IOT.....29

Figure II.10: LoRaWAN in IOT 30

Figure II.11: Message Queuing Telemetry Transport (MQTT) Protocol 31

Figure II.12: the CIA third 35

Figure III..1synoptic diagram of system 38

Figure III.2: ESP32 board. 39

Figure III.3: ESP32 function block diagram.....40

Figure III.4: The MPU6050 module. 41

Figure III.5: the Gyroscope axis (roll, pitch and yaw) 41

Figure III.6: The block diagram of the MPU6050 circuit. 42

Figure III.7: The System architecture.....43

Figure III.8: The I2C protocol.....44

Figure III.9: Arduino IDE environment.....46

Figure III.10: the installation of ESP32 in the IDE.....46

Figure III.11: Installation of ESP32 in the IDE.....47

Figure III.12: installation the library of the MPU6050. 48

Figure III.13: diagram of data collection. 51

Figure III.14: An example of .csv file.....53

Figure III.15: Accelerations and rotations for upper and lower part of body during walking.54

Figure III.16: different steps to follow to train the CNN network. 55

Figure III. 17: Importing the necessary libraries. 56

Figure III.18: processing the CSV files in each directory. 56

Figure III.19: Splitting the datasets into training and testing sets. 57

Figure III.20: The architecture of the CNN model.....57

Figure III.21: Splitting each .csv file into 4 inputs array of (1* 600). 60

Figure III.22: The new implemented architecture.....61

Figure III.23: Model Checkpoint creation. 61

Figure III.24: saving the model “.**h5**” 62

Figure III.25: Accuracy and losses graphs for training and validation. 62

Figure III.26: confusion matrix. 63

Figure III.27: the preparation model.....64

Figure III.28: real-world scenario.....65

Figure III.29: Flowchart of ESP32 code. 66

figure III.30: the results in the serial monitor. 67

List of Tables

Tableau III.1: the connection pin between the components.....44

Tableau III.2: the different position. 52

Table III.3: A comparison between two architectures. 58

Table III.4: the best result. 63

List of Abbreviations

IOT: Internet Things.

WSN: Wireless Sensor Networks.

WLAN: wireless local area network.

AI: Artificial intelligence.

ANN: Artificial Neural Networks.

CNN: Convolutional Neural Networks.

ML: machine learning.

DL: Deep Learning.

AGI: artificial general intelligence.

FEN: feed forward network traditional.

AWS: Amazon Web Services.

ITS: Intelligent transportation systems.

WPANs: wireless personal area networks.

Wi-Fi: Wireless Fidelity.

PAN: Personal area network.

LoRaWAN: Long Range Wide Area Network.

M2M: Machine to machine.

MQTT*:* Message Queue Telemetry Transport.

DVFS: Dynamic Voltage Frequency Scaling

SUs: Secondary Users.

PII: Personally identifiable information.

CR: Cognitive Radio**.**

BS: Base station.

RF: radio frequency.

CIA Triad: **C**onfidentiality, **I**ntegrity, **A**vailability.

PII: personally identifiable information

General introduction

General introduction

As individuals age, their daily living activities often undergo changes influenced by factors such as physical health, cognitive abilities, and social circumstances. Some older people live with their relatives, while others choose to live independently in their own homes. Older individuals engage in numerous daily activities, but they may also face various risks, such as falling or encountering overwhelming tasks. To address these concerns, we have undertaken a project with the aim of monitoring the daily living activities of the elderly to detect any changes in their daily routines.

With the rapid advancement of Internet of Things (IoT) technologies, monitoring and assisting the daily living activities of the elderly has become more efficient and effective. The integration of Wireless Sensor Networks (WSNs) with IoT has proven to be highly beneficial, as WSNs serve as a fundamental component of IoT systems. WSNs play a critical role in providing sensing and data collection capabilities, which are essential for the seamless operation of IoT applications.

In this project, we designed and implemented a home monitoring system to track the daily activities of senior individuals. We achieved this by recording the accelerations and rotational movements experienced by the elderly during their daily lives. Our goal was to detect any changes in their routine that could assist caregivers in providing more efficient care. To accomplish this, we utilized an ESP32 as the processing unit along with two MPU6050 sensors to detect movement patterns in both the upper and lower parts of the senior's body.

Additionally, we employed artificial intelligence techniques to predict the person's movements. These predictions were wirelessly transmitted, via WIFI, to the host device, enabling further data analysis. The transmitted information includes details such as the type of movement or the predicted position, facilitating comprehensive analysis by the host.

The first chapter provides an in-depth understanding of Artificial Intelligence (AI), including its history and applications. It also delves into the concepts of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNNs), exploring their architectures.

The second chapter covers the fundamentals of wireless sensor networks (WSN) and IoT. It includes definitions, applications, and the architecture, highlighting the different layers and components involved. Furthermore, it discusses various communication protocols used in

WSN and IoT. Additionally, this chapter emphasizes the importance of energy efficiency and addresses security and privacy challenges in WSN and IoT.

Moving on to the third chapter, we introduce the concept of our system by defining the hardware and software components used for its construction. We then delve into the process of data collection and preparation, focusing on splitting it into training, validation, and testing sets. Additionally, we explain the process of implementation of the CNN model in an ESP32 platform. Finally, we conclude this thesis with a general conclusion.

Chapter I:

An overview of Artificial Intelligence

I.1 INTRODUCTION

The objective of this chapter is to provide a comprehensive understanding of artificial intelligence (AI) and machine learning (ML), along with an exploration of their various types. Additionally, this chapter delves into the concept of Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs). Furthermore, a concise historical overview of AI is presented, followed by a categorization of different approaches to AI. Lastly, we examine the pervasive integration of AI across diverse domains in our daily lives.

I.2.Definition of artificial intelligence

Artificial Intelligence (AI) has become a buzzword in recent years, it does really mean refers to the ability of machines to perform tasks that typically require human intelligence. This includes things like recognizing speech, making decisions, and learning from experiences.

At its core, AI is all about teaching machines to think and act like humans. This involves developing algorithms that can process and analyse vast amounts of data, and then using that data to make predictions and decisions. One of the most important technologies within the field of AI is machine learning. [1]

I.3.History of AI

AI is not a modern word, is draws upon many past scientific achievements which are not mentioned here, for AI as a science in its own right has only existed since the middle of the Twentieth Century [2].

1943: Warren McCulloch and Walter Pitts was developed the first mathematical and computer model of the biological neuron (formal neuron) [3] and make the connection to propositional logic.

1950: Alan Turing was opened the doors to the field that would be called AI [4] by defined machine intelligence with the Turing test and writes about learning machines and genetic algorithms.

1956: the name Artificial Intelligence" first introduced by the American computer scientist McCarthy at the Dartmouth conference [5].

In the same year Newell and Simon present the first symbol-processing computer program called the Logic Theorist.

1966: the scientists developed algorithms which can solve the mathematical problems.

Joseph Weizenbaum created the first program which is name as ELIZA, carries out dialog with people in natural Language [6].

1972: The physician de Dombal develops an expert system for diagnosis of acute abdominal pain.

1976: Shortliffe and Buchanan develop MYCIN, an expert system for diagnosis of infectious diseases, which is capable of dealing with uncertainty.

1982: the expert system for configuring computers, saves Digital Equipment Corporation 40 million dollars per year.

1986: Renaissance of neural networks through, among others, Rumelhart, Hinton and Sejnowski . The system Nettalk learns to read texts aloud

1993: Pearl, Cheeseman, Whittaker, Spiegelhalter bring probability theory into AI with Bayesian networks.Multi-agent systems become popular.

1993: Worldwide RoboCup initiative to build soccer-playing autonomous robots.

1997: IBM's chess computer Deep Blue defeats the chess world champion Gary Kasparov.

2002: i-Robot released Roomba, an autonomous robot vacuum that cleans while avoiding obstacles.

2006: Oren Etzioni,Michele Banko, and Michael Cafarella (computer scientists), coined the term "machine reading," defining it as unsupervised autonomous understanding of text.

2009: First Google self-driving car drives on the California freeway.

2011: IBM's "Watson" beats two human champions on the television game show "Jeopardy!". Watson understands natural language and can answer difficult questions very quickly.

2015: Deep learning gives very good image classification

2016: Google DeepMind's AlphaGo, a computer program that plays the board game *Go*, defeated various (human) champions [7].

2018: A Chinese tech group called Alibaba's language-processing AI beat human intellect on a Stanford reading and comprehension test.

2020: open AI released the model GPT-3 which uses Deep Learning to create code, poetry, and other such language and writing tasks [8].

In **2023,** Microsoft Research tested the GPT-4 large-scale language model on various tasks and concluded that "it can reasonably be considered an early (but still incomplete) version of an artificial general intelligence [9] (AGI) system.

I.4.The different approaches of AI

There are four approaches of artificial intelligence based on: acting humanly, thinking humanly, thinking rationally, and acting rationally [10].

a) Acting humanly (The Turing Test approach)

Turing defined that the ability to achieve human-level performance in all cognitive tasks, which is sufficient to fool an interrogator is Intelligence of Machine The test Turing proposed is that the computer should be interrogated by a human via a teletype, and passes the test if the interrogator cannot tell if there is a computer or human at the other end.

b) Thinking humanly (The Cognitive approach)

This requires "getting inside" of the human mind to see how it works and then comparing our computer programs to this. This is what *cognitive science* is trying to do. Another way to do this is to observe a human problem solving and argue that one's programs go about problem solving in a similar way [11].

c) Thinking rationally (The laws of thought approach)

There are two main obstacles to this approach: First, it is difficult to make informal knowledge precise enough to use the logicist approach particularly when there is uncertainty in the knowledge. Second, there is a big difference between being able to solve a problem in principle and doing so in practice

d) Acting rationally (The rational agent approach)

The study of AI as rational agent design therefore has two advantages. First, it is more comprehensive approach than the ``laws of thought'', because correct inference is only a useful mechanism for achieving rationality, and not a necessary one. Second, it is more amenable to

scientific development than approaches based on human behaviour or human thought, because the standard of rationality is clearly defined and completely general [12].

This approach gives us four possible goals in artificial intelligence:

1. Systems that think like human.
2. Systems that think rationally.
3. Systems that act like humans.
4. Systems that act rationally.

I.5.Applications of AI

Artificial Intelligence has various applications in today's society. It is becoming essential for today's time because it can solve complex problems with an efficient way in multiple industries, such as Healthcare, entertainment, finance, education, etc.

AI is making our daily life more comfortable and faster. This photo will illustrate some sectors which have the application of Artificial Intelligence (Figure I.1).

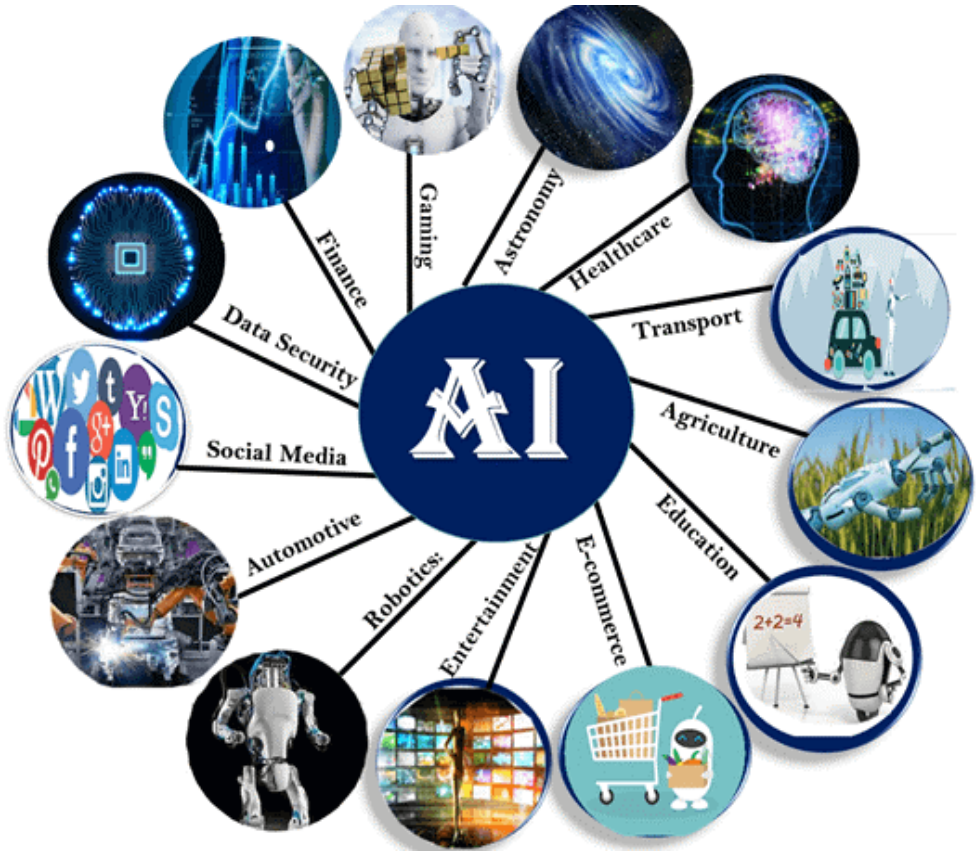


Figure I.1: the main application of AI.

I.5.1 AI in healthcare

The biggest bets are on improving patient outcomes and reducing costs. Companies are applying machine learning to make better and faster medical diagnoses than humans.
For example, there are the best-known healthcare technologies is IBM Watson. This system mines patient data and other available data sources to form a hypothesis, which it then presents with a confidence scoring schema.

I.5.2 AI in Astronomy

Artificial Intelligence can be very useful to solve complex universe problems. AI technology can be helpful for understanding the universe such as how it works, origin, etc.

I.5.3 AI in Finance

AI and finance industries are the best matches for each other. The finance industry is implementing automation, chatbot, adaptive intelligence, algorithm trading, and machine learning into financial processes.

I.5.4 AI in Data Security

The security of data is crucial for every company and cyber-attacks are growing very rapidly in the digital world. AI can be used to make your data more safe and secure. For examples there are AEG bot, AI2 Platform, are used to determine software bug and cyber-attacks in a better way.

I.5.5 AI in social media

Social Media sites such as Facebook, Twitter, and Snapchat contain billions of user profiles, which need to be stored and managed in a very efficient way. AI can organize and manage massive amounts of data. And can also analyse lots of data to identify the latest trends, hashtag, and requirement of different users.

I.5.6 AI in Travel & Transport:

AI is becoming highly demanding for travel industries. AI is capable of doing various travel related works such as from making travel arrangement to suggesting the hotels, flights, and best routes to the customers. Travel industries are using AI-powered chatbots which can make human-like interaction with customers for better and fast response.

I.5.7 AI in Automotive Industry

Some Automotive industries are using AI to provide virtual assistant to their user for better performance. Such as Tesla has introduced TeslaBot, an intelligent virtual assistant.

Various Industries are currently working for developing self-driven cars which can make your journey more safe and secure.

I.5.8 AI in Robotics

Artificial Intelligence has a remarkable role in Robotics. Usually, general robots are programmed such that they can perform some repetitive tasks, but with the help of AI, we can create intelligent robots which can perform tasks with their own experiences without pre-programmed.

the best examples for AI in robotics, recently the intelligent Humanoid robot named as Erica and Sophia has been developed which can talk and behave like humans.

I.5.9 AI in Entertainment

We are currently using some AI based applications in our daily life with some entertainment services such as Netflix or Amazon. With the help of ML/AI algorithms, these services show the recommendations for programs or shows.

I.5.10 AI in Agriculture

Agriculture is an area which requires various resources, labor, money, and time for best result. Now a day's agriculture is becoming digital, and AI is emerging in this field. Agriculture is applying AI as agriculture robotics, solid and crop monitoring, predictive analysis. AI in agriculture can be very helpful for farmers.

I.5.11 AI in E-commerce

AI is providing a competitive edge to the e-commerce industry, and it is becoming more demanding in the e-commerce business. AI is helping shoppers to discover associated products with recommended size, colour, or even brand.

I.5.12 AI in education

AI can automate grading so that the tutor can have more time to teach. AI chatbot can communicate with students as a teaching assistant.

AI in the future can be work as a personal virtual tutor for students, which will be accessible easily at any time and any place [3].

I.6.Machine Learning

I.6.1. Definition machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science. [13]

Scientists, and in order to create an artificial intelligence, use a set of different techniques grouped under the same discipline called machine learning. So, saying artificial intelligence comes back automatically to say “Machine Learning”

It involves teaching machines to learn patterns and recognize insights from data, rather than being explicitly programmed.

It uses algorithms to enable machines to improve at tasks as they are fed more data. Its purpose is to develop algorithms that can automatically learn from and make predictions or decisions based on input data, with minimal human intervention. Applications of machine learning include image and speech recognition, medical diagnosis, predictive analysis, and fraud detection.

I.6.2. The different types of machine learning

Generally, in machine learning the tasks to be solved are classified into categories. There are several methods of machine learning, but the most widely used are supervised learning, where the algorithms are based on data well-specified human inputs and outputs and unsupervised learning that does not provide the algorithm with labeled data so that it can define a structure and discover logic in the data entered.

a) Supervised learning

Supervised learning is a type of machine learning in which an algorithm learns to make predictions or decisions based on input data that is labeled or tagged with correct outputs. The algorithm is trained on a dataset that contains both input data and their corresponding outputs. During training, the algorithm tries to learn a mapping function from the input to the

output by observing the patterns and relationships between the data. Once the algorithm is trained, it can make predictions on new input data. Supervised learning is commonly used in two types of learning. The first one is Classification, where the algorithm learns to predict the categorical or discrete output variable of a dataset based on the input variables. the dataset includes a target variable or label that represents the class or category of each observation. The goal is to train a model that can accurately predict the class of a new observation based on its features.

The second one is regression used to predict a continuous output variable based on one or more input variables. Supervised machine learning helps to solve various types of real-world computation problems [14].

b) Unsupervised learning

This branch of machine learning consists of finding interesting transformations of the input data without the help of any targets [15]. where there are an algorithm learns to identify patterns or structures in the data on its own and groups the data into clusters based on their similarities.

The main objective of unsupervised learning is to identify hidden patterns in the data and gain insight into the data without any specific goal or objective in mind. Some of the common techniques used in unsupervised learning include clustering were separating the similar objects or data points into groups/clusters so that data points within a cluster are more similar to each other than to data points in other clusters., dimensionality reduction, and association rule mining. Unsupervised learning is often used in data mining, natural language processing, and image recognition.

c) Reinforcement learning

This branch of machine learning has only recently begun. In the context of reinforcement learning, autonomous agents must learn to perform tasks through trial and error without any guidance human operator. DeepMind demonstrated that a deep learning-based reinforcement learning system can learn to play an Atari video game and achieve human-level performance on a number of tasks. Deep learning has also greatly improved the performance of reinforcement learning in robots [16].

The next figure illustrates the main types of machine learning [17].

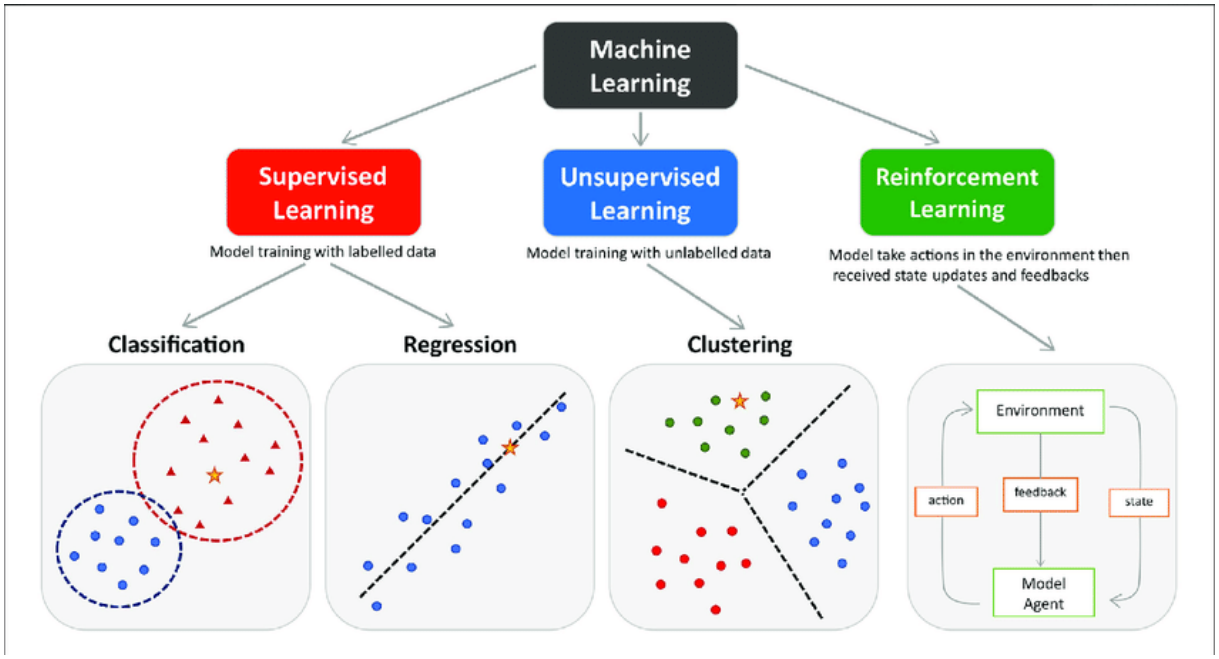


Figure I.2: Main types of machine learning.

I.7.Deep Learning

deep learning is also referred to as Artificial Neural Networks (ANNs).

It is subset of machine learning which is a subset of AI as show in figure.1.3. It involves training complex neural networks, typically with multiple layers, to learn from data and make predictions or decisions on new data [16].

Deep learning has been successfully applied in a wide range of applications such as computer vision, natural language processing, speech recognition, and even game playing.

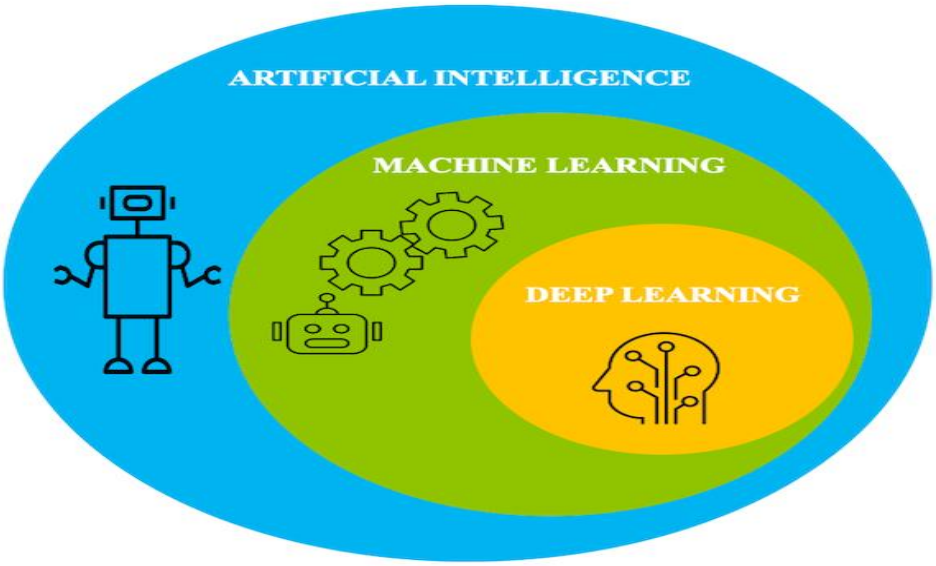


Figure I.3: Deep learning is subset of machine learning.

I.7.1. Artificial Neural Networks (ANNs)

a) Definition

Artificial neural networks (ANN) are a type of machine learning algorithm modelled after the structure and function of the human brain. They consist of interconnected nodes or "neurons" that work together to perform a specific task such as classification or prediction.

ANNs can be designed to learn and improve from experience, allowing them to adjust and modify their behaviour over time. They are useful for tasks such as image and speech recognition, natural language processing, and predictive modelling.

ANNs work by receiving input data, processing it through a series of connected nodes, and producing an output based on the specified task. The network is trained by providing it with examples of input/output pairs, allowing it to learn and improve its performance.

Artificial neural networks are a powerful tool in machine learning and have numerous applications in various industries.

b) Architecture of Artificial Neural Network

To understand the architecture of an artificial neural network, that need to understand what a typical neural network contains. In order to describe a typical neural network, it contains a large number of artificial neurons (that is why it is called an artificial neural network) which are termed units arranged in a series of layers. Figure 1.4 allows us take a look at the different kinds of layers available in an artificial neural network [18].

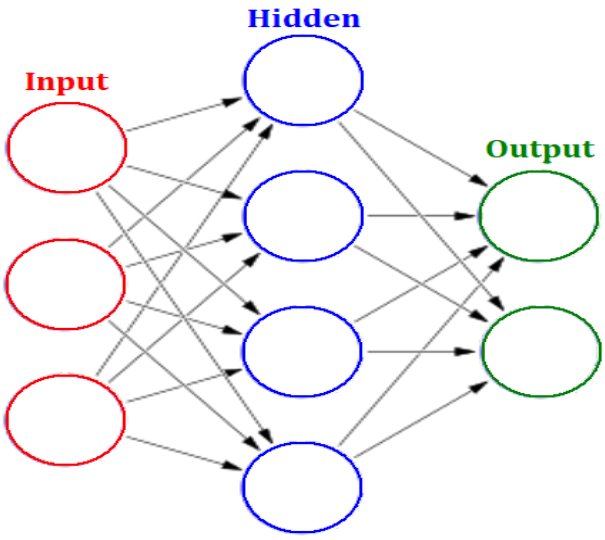


Figure I.4: The different layers of artificial neural network.

- **Input layer:**

The Inputs layer contain those artificial neurons (termed as units) which are to receive input from the outside world. This is where the actual learning on the network happens, or recognition happens else it will process.

- ***Output layer***

The output layers contain units that respond to the information that is fed into the system and also whether it learned any task or not.

- **Hidden layer**

The hidden layers are mentioned hidden in between input layers and the output layers. The only job of a hidden layer is to transform the input into something meaningful that the output layer. units can use in some way.

Most of the artificial neural networks are all interconnected, which means that each of the hidden layers is individually connected to the neurons in its input layer and also to its output layer leaving nothing to hang in the air. This makes it possible for a complete learning process and also learning occurs to the maximum when the weights inside the artificial neural network get updated after each iteration. [19]

c) The functioning of artificial neural networks

Artificial Neural Networks can be best viewed as weighted directed graphs, where the nodes are formed by the artificial neurons and the connection between the neuron outputs and neuron inputs can be represented by the directed edges with weights. The Artificial Neural Network receives the input signal from the external world in the form of a pattern and image in the form of a vector. These inputs are then mathematically designated by the notations x(n) for every ‘n’ number of inputs.

Each of the input is then multiplied by its corresponding weights (these weights are the details used by the artificial neural networks to solve a certain problems). In general terms, these weights typically represent the strength of the interconnection amongst neurons inside the artificial neural network. All the weighted inputs are summed up inside the computing unit (yet another artificial neuron).

If the weighted sum equates to zero, a bias is added to make the output non-zero or else to scale up to the system's response. Bias has the weight and the input to it is always equal to 1. Here the sum of weighted inputs can be in the range of 0 to positive infinity. To keep the response in the limits of the desired value, a certain threshold value is benchmarked. And then the sum of weighted inputs is passed through the activation function.

The activation function, in general, is the set of transfer functions used to get the desired output of it. There are various flavours of the activation function, but mainly either linear or non-linear sets of functions. Some of the most commonly used set of activation functions are the Binary, Sigmoidal (linear) and Tan hyperbolic sigmoidal (non-linear) activation functions. Figure 1.5 illustrates an artificial neural networks Works

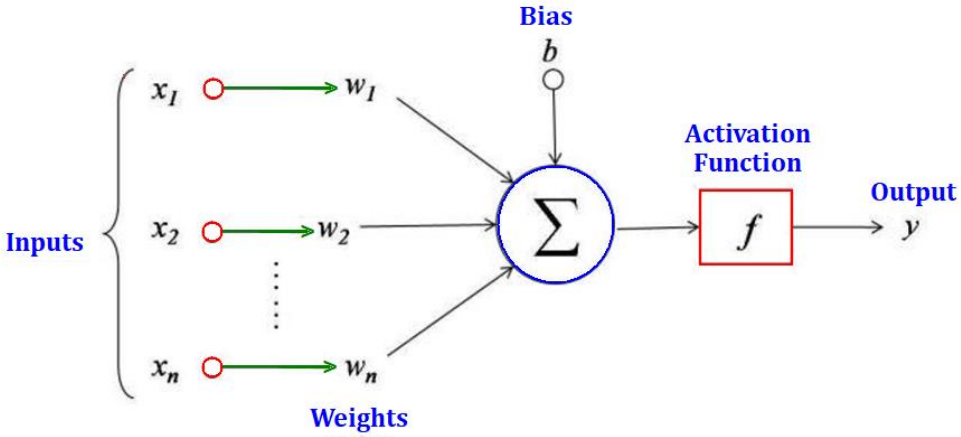


Figure I.5: Artificial neural networks Works

I.7.2 Convolution neural networks

A convolutional neural network or CNN is an extension of feed forward network traditional (FFN) within the framework of the inspiration of biological factors.

It is a type of deep learning algorithm commonly used in image and video processing applications.[20]

It is a multi-layer neural network that uses mathematical algorithms to learn pattern recognition and feature detection from pixel values of images or videos. CNNs are widely used in image classification, object detection, facial recognition, and autonomous vehicles.

CNNs work by applying convolutional layers that extract specific features from the input data, followed by pooling layers that aggregate the extracted features, and finally fully connected layers that classify the data.

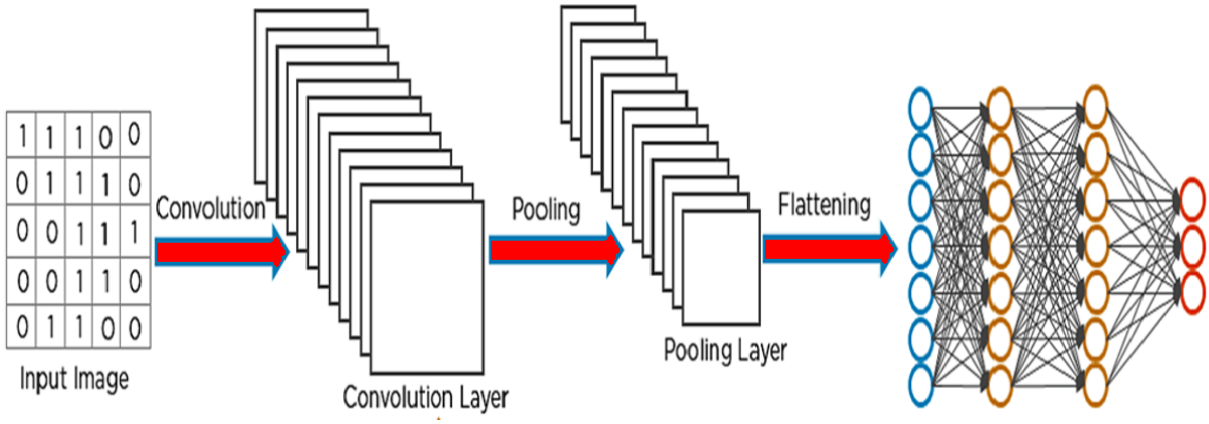


Figure I.6: Convolution neural network

a) convolutional layers

The objective of the convolution is to extract the high-level features.

Convolutional layers are a fundamental building block of convolutional neural networks (CNNs), which are the go-to deep learning model for image and video processing tasks.

Convolutional layers consist of a set of learnable filters (also called kernels) that slide over the input feature map to produce a set of output feature maps, with each filter scanning the input at a different location to extract a local pattern or feature. Figurre1.7 illustrates One step in the Convolution Process [21].

During training, the filters are adjusted to minimize the objective function of the model and improve its ability to recognize patterns in images or videos.

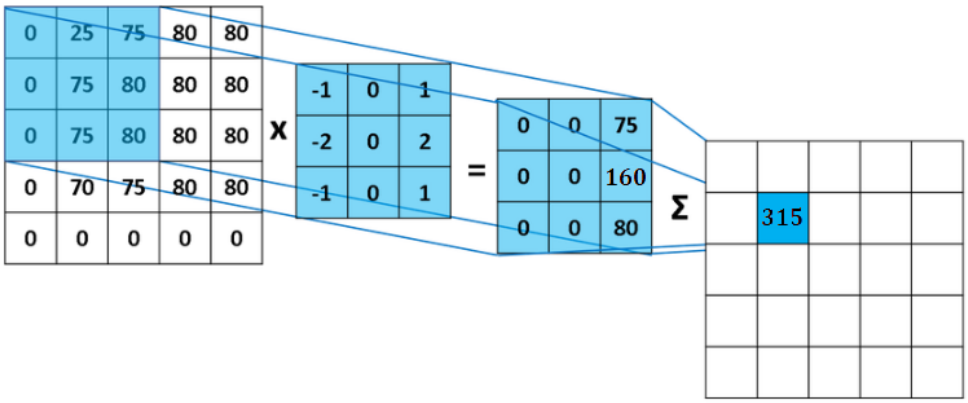


Figure I.7: One step in the Convolution Process.

Convolutional layers are typically stacked one after the other, with each subsequent layer processing the outputs of the previous layer to extract more complex and abstract features. The final output can then be used for classification, segmentation, detection, or other downstream tasks.

Notice that there is a border of empty values around the convolved image. This is because the result of convolution is placed at the centre of the kernel. To deal with this, a process called 'padding' or more commonly 'zero-padding' is used. This simply means that a border of zeros is placed around the original image to make it a pixel wider all around. The convolution is then done as normal, but the convolution result will now produce an image that is of equal size to the original. The figure I.8 illstarate the convolutional layers works With and without zero padding [21].

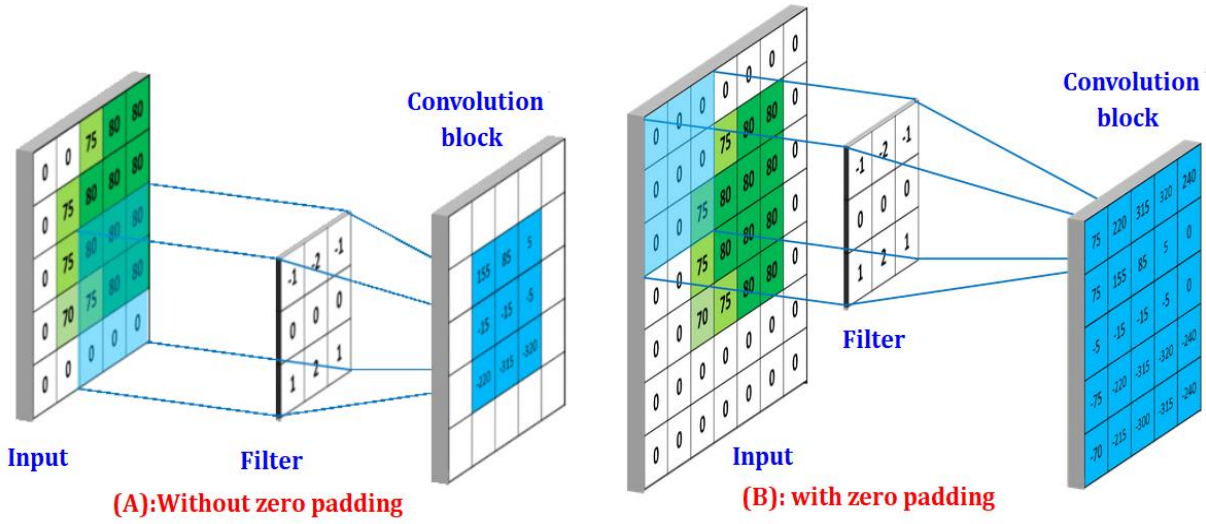


Figure I.8: Convolutional layers works With and without zero padding.

b) pooling layers

Pooling layers, also known as down sampling layers, are a common component used in Convolutional Neural Networks (CNNs). After a convolutional layer, a pooling layer typically follows to reduce the spatial size of the feature maps. The main functions of pooling layers are:

1. To reduce the spatial dimensionality of the feature maps (width and height) to decrease the computational requirements of the neural network.

2. To make the network more robust to small variations in the input image by summarizing the features maps, which can also help reduce overfitting.

Figure 1.9 illustrates how pooling layers work [21].

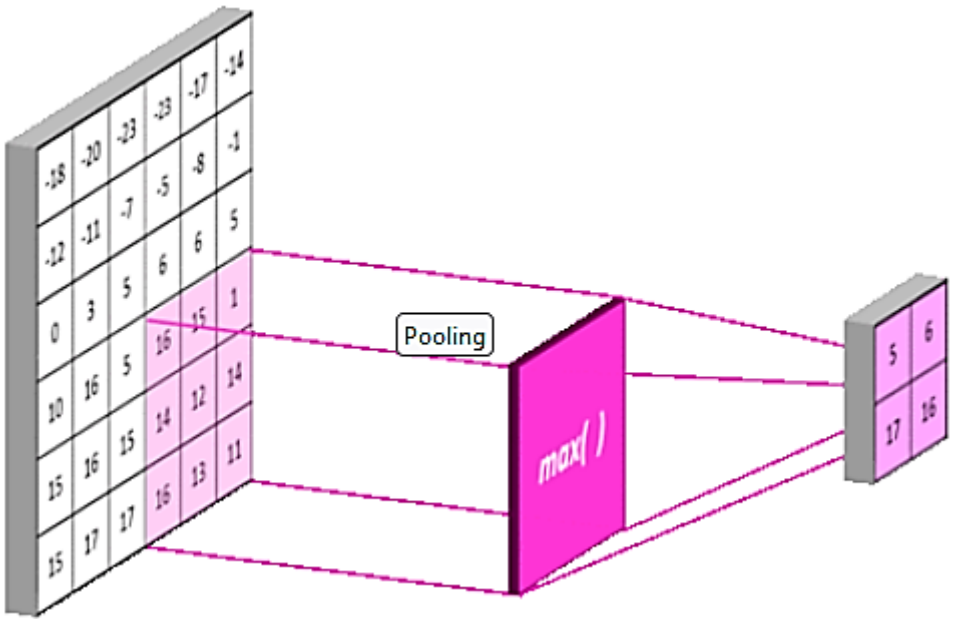


Figure I.9: The pooling layers.

There are several types of pooling layers, such as max pooling, average pooling, and global pooling.

Max pooling is the most commonly used method, where the maximum value within a neighborhood (usually a 2x2 or 3x3 window) is taken as the output for that region.

Average pooling computes the average value of the features in a neighborhood, while global pooling computes the average or maximum of all the features within a feature map.

The pooling layers are an important tool in CNNs for improving computational efficiency, reducing overfitting, and improving the translational invariance of the network to small image shifts or distortions. Figure 1.10 illustrate two types of pooling layers.

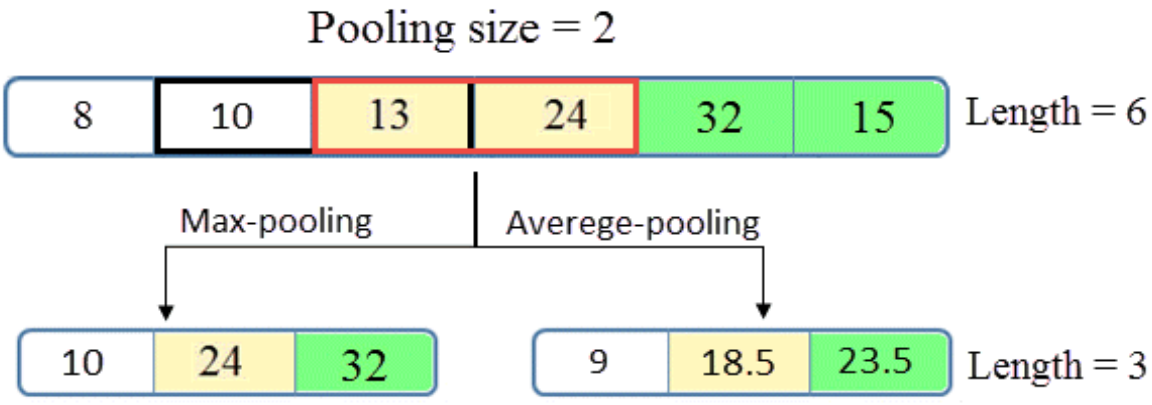


Figure I.10: The types of pooling layers.

c) Fully-connected layers

Fully-connected layers, also known as linear layers, connect every input neuron to every output neuron and are commonly used in neural networks as shows a Figure I.11 [22].

Fully-connected layers in CNN (Convolutional Neural Network) are the layers where all the neurons in one layer are connected to all the neurons in the next layer. These layers are also called dense layers. In a typical CNN, a series of convolutional and pooling layers extract the features from the input image and generate a 3D tensor. The output of these layers is then flattened into a vector, which is fed into one or more fully-connected layers to perform the classification task.

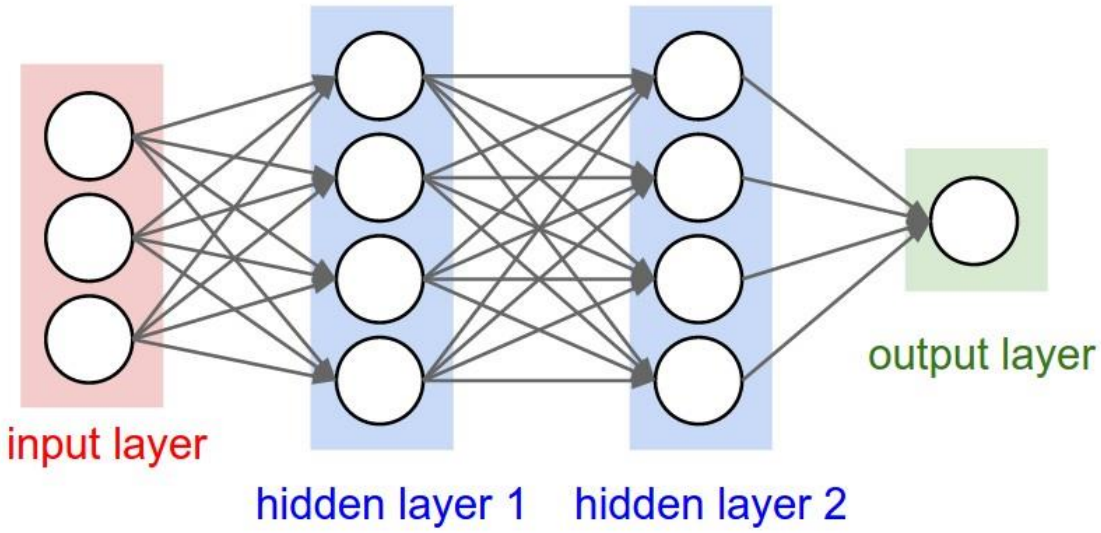


Figure I.11: Fully-connected layers.

Fully-connected layers are computationally expensive and require a lot of memory. To reduce the number of parameters, techniques such as dropout, weight regularization, and batch normalization can be used. In summary, fully-connected layers are an important part of CNNs for performing the final classification task.

d) Output Layer:

Depending on the purpose of your CNN, the output layer will be slightly different. In general, the output layer consists of a number of nodes which have a high value if they are 'true' or activated. In fact, the FC layer and the output layer can be considered as a traditional NN where we also usually include a SoftMax activation function. Some output layers are probabilities and as such will sum to 1, whilst others will just achieve a value which could be a pixel intensity in the range 0-255. The output can also consist of a single node if we're doing regression or

deciding if an image belong to a specific class or not e.g., diseased or healthy. Commonly, however, even binary classification is proposed with 2 nodes in the output and trained with labels that are 'one-hot' encoded i.e. [1,0] for class 0 and [0,1] for class 1.[23]

I.8. Conclusion

This chapter delved into the domain of artificial intelligence (AI), exploring its definitions, historical development, diverse applications, and key techniques. We embarked on a journey through the evolution of AI, from its early theoretical foundations to the transformative breakthroughs that have shaped our modern world.

We first examined the different definitions of AI, recognizing its overarching goal of creating intelligent systems that can mimic human cognitive abilities.

The chapter then delved into the historical timeline of AI, tracing its origins back to the Dartmouth Workshop in 1956 and following its ebbs and flows through periods of optimism, known as AI summers, and periods of skepticism, the AI winters. Despite these ups and downs, AI has persevered and experienced a resurgence in recent years, thanks to advancements in computing power, data availability, and algorithmic improvements.

We also delved into the vast array of AI applications across various domains, including healthcare, finance, transportation, and entertainment. From medical diagnosis and autonomous vehicles to natural language processing and virtual assistants, AI has revolutionized countless industries and continues to reshape our daily lives.

A significant portion of this chapter was dedicated to exploring the fundamental techniques that underpin AI. We examined machine learning, a subfield of AI that empowers systems to learn from data and make predictions or decisions without explicit programming. In particular, we delved into convolutional neural networks (CNNs), a powerful deep learning architecture specifically designed for visual processing tasks.

In the upcoming chapter, we will embark on a new exploration, uncovering the remarkable world of interconnected devices, where sensors and actuators collaborate. We will delve into the architecture, protocols, and applications of WSN and IoT, shedding light on how these interconnected systems are transforming industries, cities, and our everyday lives.

Chapter II:

Wireless Sensor Networks and IoT

II.1 Introduction

Wireless Sensor Networks (WSNs) and Internet of Things (IoT) are both related to the deployment of wireless sensors and communication devices for collecting and transmitting data. WSN and IOT are typically used in applications such as environmental monitoring, industrial automation, healthcare, smart homes and cities. In this chapter we will introduce the essential concepts of WSN and IoT. We will first define the term of IoT and WSN, describe a WSN and IOT Architecture and discuss the different components of them. Then we will cover the different communication protocols used in WSN and IoT, such as ZigBee, Bluetooth Low Energy (BLE), Wi-Fi, LoRa WAN, and MQTT. Also, we will describe the importance of energy efficiency in wireless sensor networks and IoT, and the techniques used to extend the battery life of sensors nodes.

This chapter also covers security and privacy challenges in wireless sensor networks, and the different application in various domains of our life.

II.2 Wireless Sensor Networks (WSN)

II.2.1 Definition

Wireless Sensor Networks (WSNs) have become a popular technology for data collection in recent years. They are composed of small, low-power devices called sensor nodes that can communicate wirelessly with each other to collect and transmit data. The collected data is then transmitted wirelessly to a central base station or data server for further processing and analysis.

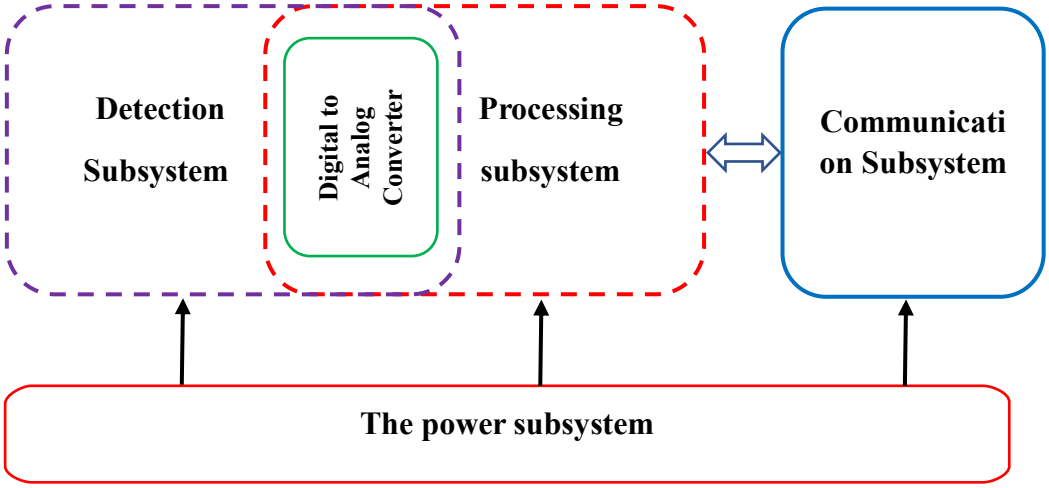


Figure II.1: Architecture of a wireless sensor node.

As shown in figure II.1 The node includes basicaly sensing, processing, communication, and power subsystems.

II.2.2 WSN Architecture

The wireless sensor network architecture is built with nodes that are used to observe the surroundings like temperature, humidity, pressure, position, etc. [24]

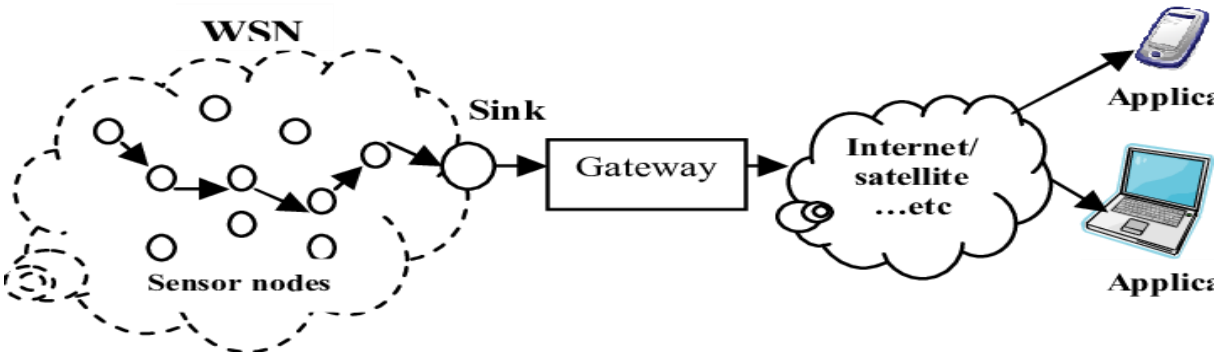


Figure II.2: WSN architecture.

II.2.3 The different components of WSN

WSN will have three main components distributed sensor nodes, sink nodes (Base station or Gateway) and Software WSNs, as shown in Figure II.2, are composed of a number of sensor nodes, which are densely deployed either inside a physical phenomenon or very close to it.

a) **The sensor nodes:** Wireless sensor networks (WSNs) are composed of a huge number of sensor nodes. They are transceivers usually scattered in a sensor field where each of them has the capability to collect data and route data back to the sink/gateway and the end-users by a multi-hop infrastructure less architecture through the sink. They use their processing capabilities to locally carry out simple computations and transmit only the required and partially processed data.

b) **Gateway** is located at the boundary of the monitoring coverage area This gateway serves as a connection between the micro-sensor network and external entities, such as the internet or other wireless sensor networks.

c) **Software WSNs:** WSNS stands for Wireless Sensor Networks Software. It refers to the software systems that are used to manage and control wireless sensor networks. These software systems include applications, protocols, middleware, and operating systems that are designed to enable communication, data exchange, and processing between

sensors and the network infrastructure. WSNs software provides various functionalities such as sensing, transmission, data processing, and control of sensors and the network. These software systems are critical components of any wireless sensor network as they enable the efficient and reliable operation of the network.

II.3 Internet of Things (IOT)

II.3.1 definition

The Internet of Things (IoT) refers to the network of physical objects, devices, vehicles, buildings, and other items that are connected to the internet and can exchange data with each other. These objects are embedded with sensors, software, and network connectivity that enable them to collect and exchange information, and to be remotely monitored and controlled. The IoT allows for real-time interactions between humans and machines. The IoT allows for the objects to communicate with each other and with other devices, such as computers and smartphones. This communication can happen through a variety of means, including Wi-Fi, Bluetooth, and cellular data. The IoT has the potential to revolutionize the way we live, work, and play.

II.3.2 IoT Architecture

The term "IoT architecture" refers to the various methods used in constructing IoT devices to match the needs of the users. can comprise up to seven layers, which are known as the perception, transport, edge, processing, application, business, and security layers.

a) The main layers in IoT architecture:

- **The perception:** layers of an IoT system architecture, also known as Things layer, consists of multiple elements – sensors connected to the internet.[24]
- **Network Layer:** This layer is responsible for connecting the physical layer with the application layer. It encompasses components of the network infrastructure, including routers, gateways, and satellites, that enable communication and data transfer between devices
- **Processing layer:** The collection, analysis, and transmission of data from IoT devices to the cloud or other applications is the responsibility of the processing layer. This layer is composed of hardware and software components that are involved in the collection, processing, analysis, and transmission of data. The edge devices that collect data, as

well as the platforms and software that facilitate data processing and analytics, are both included in the processing layer.

- **The application layer:** of an IoT system architecture involves decoding promising patterns in IoT data and compiling them into summaries that are easy for humans to understand, such as graphs and tables. Programs for device control and monitoring, as well as process control software, are typical examples of the application layer of IoT architecture.

b) The different components in IOT

- **Sensors:** The field of IoT involves the utilization of sensors, which are electronic devices, to detect and gather data from the physical world. These sensors are responsible for collecting data on various physical parameters such as temperature, humidity, pressure, motion, light, sound, etc. These sensors send the data they collect to a gateway device for further processing.
- **Gateways:** Gateways are devices that connect IoT sensors to the interne. they act as a bridge between the sensor and the cloud platform, collecting data from the sensors and transmitting it to the cloud. Furthermore, gateways often possess supplementary abilities such as data filtering and analysis, data storage and protocol translation.
- **Cloud platform**: The processing and analysis of IoT data are carried out on cloud platforms. These platforms are designed to offer tools and services for the storage, processing, and analysis of the IoT data. Some of the popular cloud platforms for IoT are: Amazon Web Services (AWS) IoT Platform, Microsoft Azure IoT, Google IoT, IBM Watson IoT, Cisco IoT Cloud Connect, Things Board Open-Source IoT Platform and Oracle IoT Intelligent Applications. these platforms provide security features, scalability, and data visualization to their users.
- **Applications** The application in IoT refer to software programs that use the data collected from sensors to provide valuable insights and enable intelligent decision-making

II.4 The different applications of wireless sensor networks and IoT

II.4.1 Healthcare:

The integration of WSNs and IoT in healthcare systems allows for real-time monitoring, data collection, and analysis. There are some ways in which healthcare can benefit from WSNs and IoT the figure II.3 illustrate the application of WSN in healthcare [25].

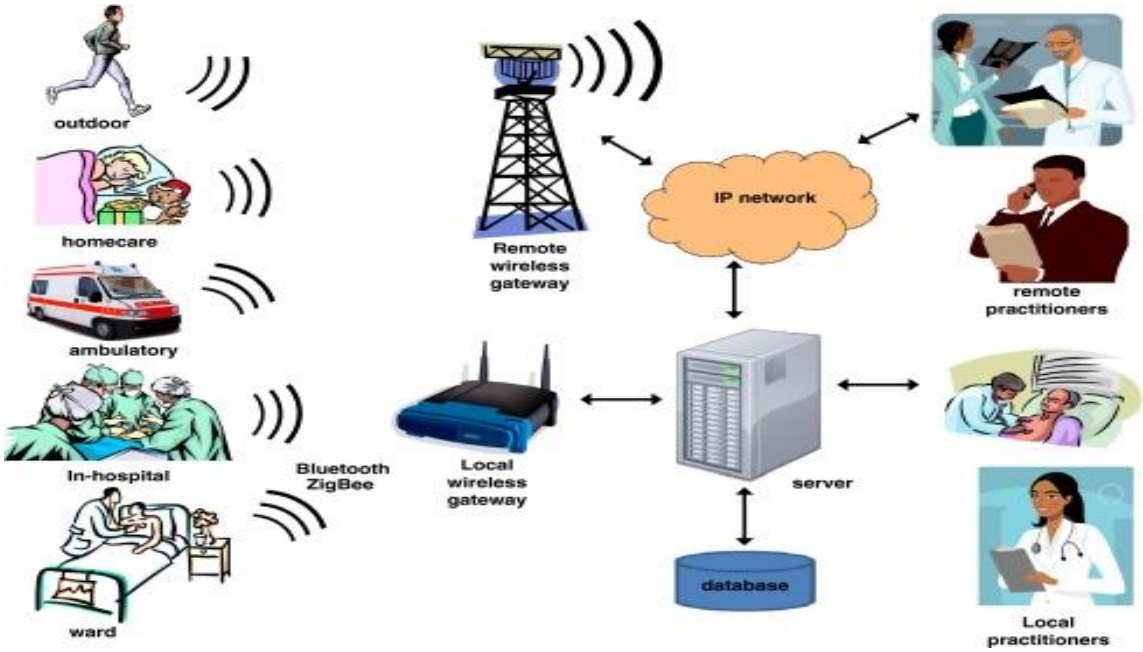


Figure II.3: WSN in healthcare application.

a) **Remote patient monitoring:** WSN and IoT sensors can be used to monitor patients' vital signs, including heart rate, blood pressure, temperature, and other biometric information. These sensors can then transmit the data in real-time to healthcare providers, enabling them to monitor the patient's condition and provide timely interventions if necessary.

b) **Ambient assisted living:** WSN and IoT technologies can be used to create smart homes and assisted living environments that enable elderly or disabled patients to live independently. The sensors in such environments can detect falls, track movements, and provide real-time alerts to caregivers or family members in the case of an emergency.

c) **Medication management**: WSN and IoT sensors can help patients manage their medications by providing reminders when it is time to take their medicine. This can help improve medication adherence and reduce the risk of medication errors.

d) **Infection control**: WSN and IoT sensors can be used to monitor and track infections within healthcare facilities. This can help identify outbreaks early and enable healthcare providers to take timely measures to prevent their spread.

II.4.2 Agriculture

a) **Soil Monitoring:** Optimizing crops through frequent monitoring of soil conditions is pivotal to a successful harvest. Data gathered from soil sensors that track crucial metrics such as moisture levels, temperatures, and pH can provide insightful information for irrigation and fertilization adjustments as shows the figure II.4. [26]



Figure II.4: Soil quality Monitoring.

b) **Crop Monitoring**: The use of IoT sensors enables the monitoring of crop growth, detection of pests and diseases, and the assessment of plant health in the field of crop monitoring.

c) **Management:** Water is a valuable resource that farmers must handle with care. To ensure efficient usage of water, IoT sensors are utilized to monitor water consumption, detect leaks, and optimize irrigation schedules by taking into account weather patterns and soil moisture levels

d) **Livestock Monitoring**: Potential issues with livestock health can be identified and their activity levels tracked using sensors for livestock monitoring. This allows for the monitoring of livestock health and behaviour.

e) **Precision Agriculture**: The integration of WSNs and IoT in agricultural operations, referred to as Precision Agriculture, allows farmers to optimize their resources by using the right-amount of inputs such as fertilizers and pesticides leading to higher yields and improved profits.

II.4.3 Transportation

a) **Smart Traffic Management**: The implementation of Smart Traffic Management involves the use of WSNs and IoT to accurately detect and monitor traffic conditions

in real-time. By doing so, traffic management systems can effectively enhance their ability to optimize traffic flow and minimize traffic congestion [27].

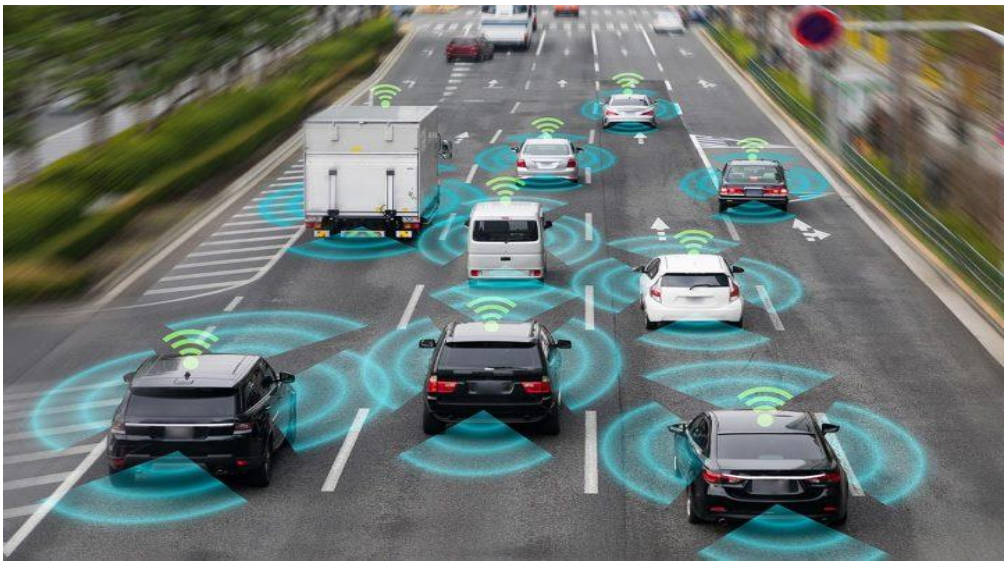


Figure II.5: WSNs and IoT in smart Traffic Management.

b) **intelligent transportation systems (ITS):** or intelligent transportation systems, can benefit from the utilization of WSNs and IoT to gather real-time data on road and weather conditions, traffic signals, and emergency services.

c) **smart parking:** By utilizing IoT sensors, smart parking systems can promptly detect the availability of parking spaces, and notify drivers in real-time. This approach helps decrease traffic congestion and enhances the overall efficiency of the parking system.

d) **Condition monitoring:** By attaching sensors to transportation infrastructure, such as bridges and tunnels, it is possible to monitor their condition and detect potential faults, which in turn enables early repairs and maintenance.

e) **Fleet Management:** The management of fleets can be greatly improved by utilizing WSNs and IoT devices. These technologies can enable the real-time tracking of vehicles, facilitate fuel consumption monitoring, and streamline routing. The outcome of these benefits is a reduction in costs and an enhancement of overall operational efficiency.

II.4.4 smart cities:

In the context of IoT and WSN, a smart city refers to a city that uses sensors, devices, and data analysis to improve the quality of life for citizens and enhance the efficiency of urban services. IoT and WSN technologies have a significant role to play in enabling smart cities by providing

real-time data and information about various aspects of the city. We have previously mentioned some various application of IOT and WSN in smart cities like healthcare, agriculture, transportation [28].

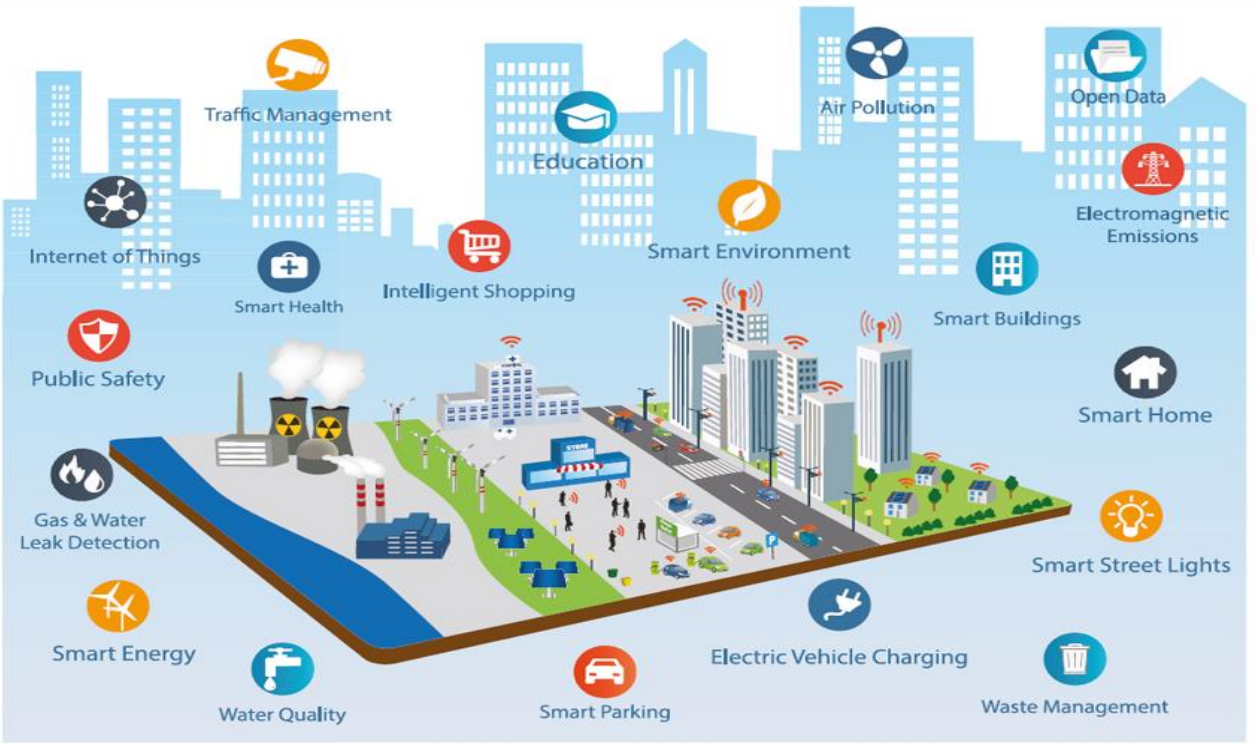


Figure II.6: WSN and IOT in smart cities.

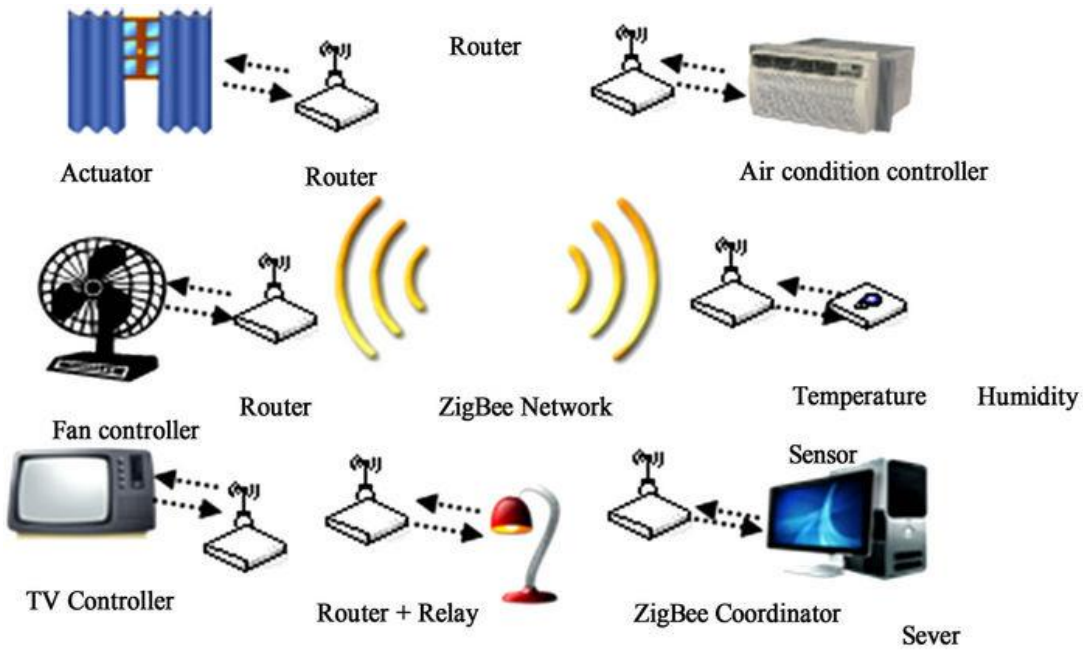
II.5 The different communication protocols used in WSN and IOT

WSNs and IoT have overlapping communication requirements, and several protocols are commonly used in both domains. These are some of the widely used communication protocols in both WSNs and IoT.

II.5.1 ZigBee

ZigBee is a specification for a suite of high-level communication protocols used to create wireless networks built from small, low-power digital radios. ZigBee is based on an IEEE 802.15.4 standard. It is a simple, efficient, reliable, and low cost, low-power standard of wireless technology. Though its low power consumption limits transmission distances to 10-100 meters line-of-sight, depending on power output and environmental characteristics ZigBee is typically used in low data rate applications that require long battery life and secure networking) ZigBee has a defined rate of 250 Kbit/s, best suited for intermittent data

transmissions from a sensor or input device [29]. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or Wi-Fi [30]. the figureII.7 illustrate the Wireless sensor and actor networks based on ZigBee protocol [31].



FigureII.7: Wireless sensor and actor networks based on ZigBee protocol.

In IOT, ZigBee is majorly used in industrial settings and home automation applications, and it's one of the most popular mesh protocols in IoT environments.

II.5.2 Bluetooth

Bluetooth is a short-range wireless technology that uses short-wavelength, ultrahigh-frequency radio waves. It had most commonly been used for audio streaming, but it has also become a significant enabler of wireless and connected devices [32].

Bluetooth Low Energy is a power-conserving variant of Bluetooth personal area network (PAN) technology, designed for use by Internet-connected machines and appliances as show figureII.8. Also marketed as Bluetooth Smart, Bluetooth LE was introduced in the Bluetooth 4.0 specification as an alternative to Bluetooth Classic [33].

The figure II.8 show Bluetooth Low Energy Applications in IOT [34].

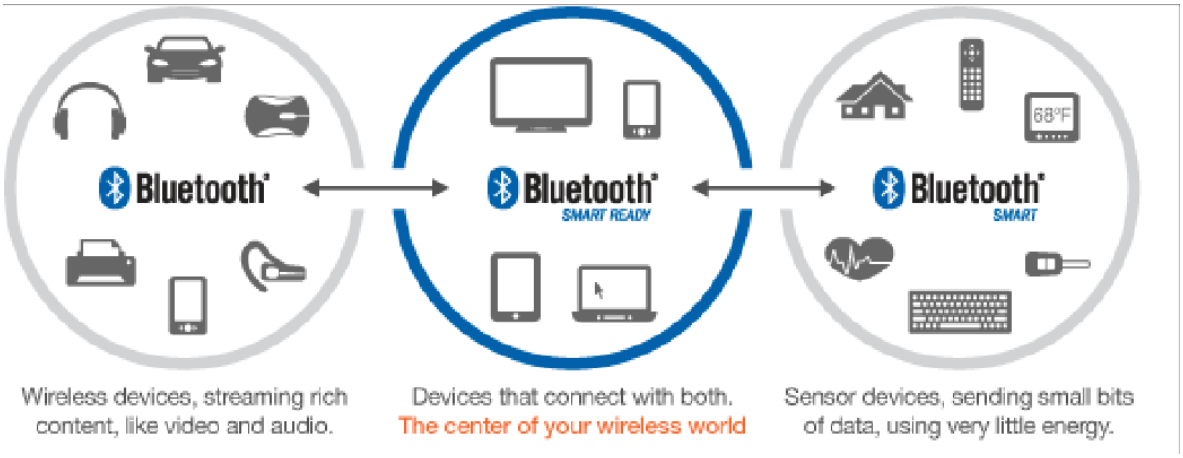


Figure II.8: Bluetooth Low Energy Applications in IOT.

II.5.3 Wi-Fi

Wi-Fi (Wireless Fidelity) is the most popular IOT communication protocols for wireless local area network (WLAN) that utilizes the IEEE 802.11 standard through 2.4 GHz UHF and 5 GHz frequencies. Wi-Fi provides Internet access to devices that are within the range of about 20 - 40 meters from the source. It has a data rate up to 600 Mbps maximum. In embedded systems, **ESP** series controllers from Espressif are popular for building IoT based Applications. **ESP32** and **ESP8266** are the most commonly use Wi-Fi modules for embedded applications [35]. the Figure II.9 show Wi-Fi application in IOT [36].

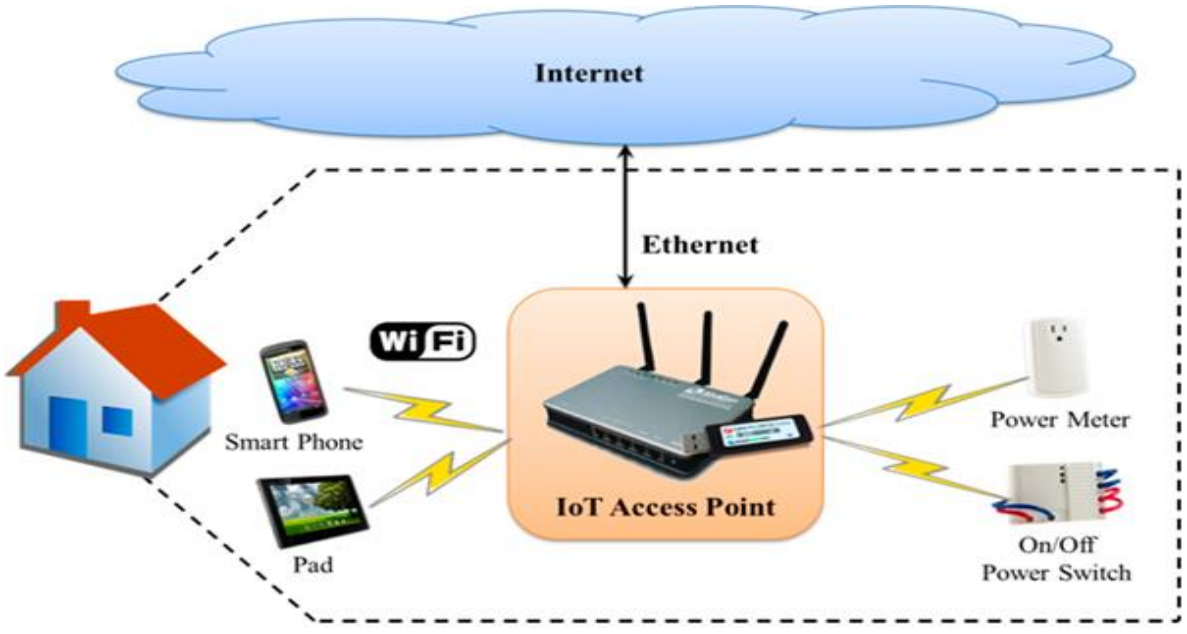


Figure II.9: Wi-Fi application in IOT.

II.5.4 LoRaWAN

LoRaWAN:(LongRange Wide Area Network) constitutes one of the leading LPWAN technologies and has received considerable attention in industrial and research communities.it has applications for long distances and is designed to provide low-power for communication in IoT, M2M applications**.** It has a capacity of connecting millions of devices with data rates ranging from 0.3 kbps to 50 kbps.it used for IoT devices like smart agriculture, smart city, and industrial automation as show figureII.10 [37;38].

.

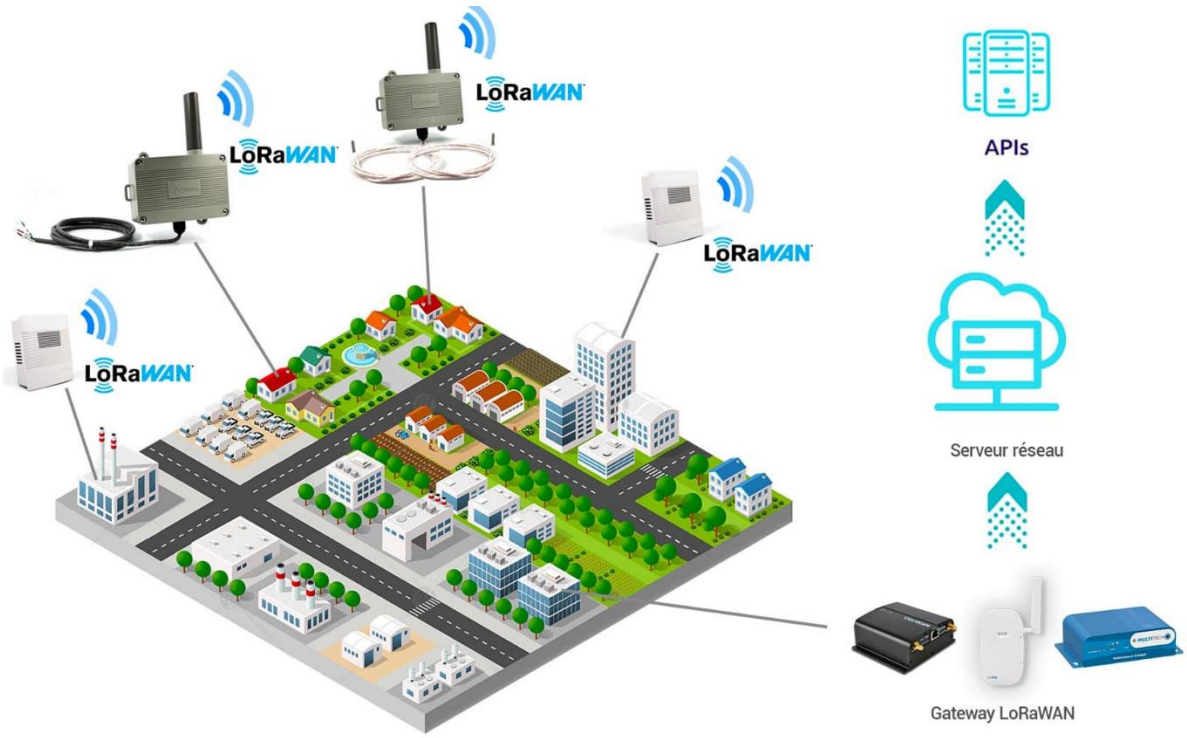


Figure II.10: LoRaWAN in IOT.

II.5.5 MQTT

MQTT **(**Message Queue Telemetry Transport) is a standard messaging protocol for the Internet of Things (IoT). it is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code. MQTT, it used for IoT devices that require instantaneous data transmission, such as in industrial automation systems (figure II.11) [39].

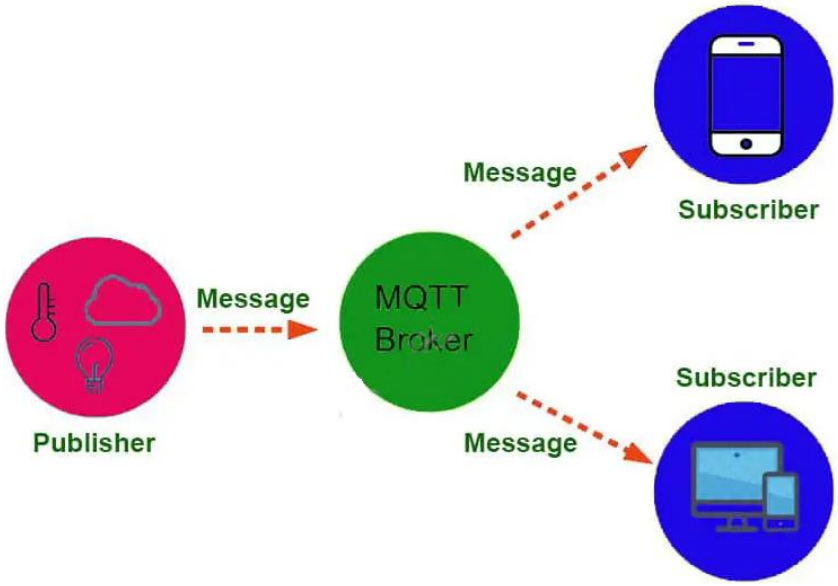


Figure II.11: Message Queuing Telemetry Transport (MQTT) Protocol.

II.6 Energy Efficiency

Wireless Sensor Networks (WSN) and the Internet of Things (IoT) are two technologies that enable a wide range of applications in various domains. One of the critical challenges for deploying and sustaining WSN and IoT applications is energy efficiency which is so important for the deployment and success of the IOT and WSN.

There are some reasons why energy efficiency is crucial in WSNs and IoT

- **Extending battery life:** Wireless sensor nodes are battery-powered, and it can be challenging to replace or recharge the batteries in remote locations. Energy-efficient devices and protocols help to minimize power consumption, thus extending their operational life.
- **Reducing operational costs**: Energy efficiency reduces the amount of power that devices consume, leading to reduced energy costs and long-term savings.
- **Enhancing scalability**: IoT devices and wireless sensor networks are designed to operate on a massive scale. However, energy-efficient protocols and devices can enhance scalability by reducing power consumption, enabling the deployment of more devices in the network.

II.6.1. Techniques used to extend the battery life of sensors nodes:

Energy efficiency is an essential aspect of (WSNs) and IoT devices. These networks and devices rely on battery power, which means that energy consumption must be carefully managed to extend battery life.

Implementing power management techniques such as dynamic voltage scaling, duty cycling, and sleep scheduling can help reduce the energy consumption of the sensor nodes.

- **Dynamic Voltage Frequency Scaling (DVFS)** presents one of the most energy-efficient management techniques by solving both challenges of on-demand performance and energy reliability. Generally, it is introduced as a framework that enables the change of the frequency and operating voltage of the processor based on system performance requirements at a certain point in time.[40].
- **duty cycling:** duty cycle becomes the basic mechanism of energy conservation in IoT networks. Duty Cycle involves the fraction of time to keep the node in sleep and active state. It is the ratio between the wake-up time length in a predefined period and the total length of the period [41]. Keeping in view the significance of duty cycling techniques, several techniques have been emerged from the last few decades. Duty cycling depends upon certain factors such as synchronization of nodes, hardware/software complexities, network area, topology, and latency [42].
- **Sleep scheduling** in randomly deployed WSNs is an effective method to reduce the energy consumption of sensor nodes, which can extend the network lifetime while ensuring network coverage.[43]
- **Energy Harvesting**: The goal of using Energy Harvesting wireless IoT sensors is to give these sensors a nearly unlimited source, such as solar, thermal, or vibration, to power the device without having to wire it to the main power [44]
- **Cognitive Radio (CR)** is a new paradigm that has the potential to efficiently utilize the unused licensed spectrum bands, also known as spectrum holes, by dynamically allocating the spectrum holes to unlicensed users referred to as Secondary Users (SUs) without any harmful interference with PUs' transmissions. Therefore, the main motivation for CR is dynamic access to temporal and spatial spectrum holes [45]
- **Data redaction**: Data redaction is the process of removing or blurring sensitive information in the data, such as personally identifiable information (PII) or confidential data, before it is transmitted over the network. By doing so, the amount of data that needs

to be transmitted is reduced, which in turn conserves energy and extends the battery life of the IoT devices and WSN sensors.

- **Data compression**: The quantity of energy consumed by a transceiver in a WSN strongly depends on the size of the transmitted data packets and the transmission distance. Data compression focusses on reducing the size of the data sent. In this technique, information is encoded at the source nodes and decoded at the BS [45].
- **Data aggregation** is a data reduction scheme for minimizing communication overhead. It is an approach that eliminates redundant data to reduce energy consumption and prolong the lifetime of WSNs [46]. data aggregation saves energy by avoiding transmission of the same data by several nodes via multiple routes. [47]
- **Data prediction** is a strategy that conserves energy in WSNs by reducing the number of data transmissions [48]. In this strategy, algorithms are utilized to approximate future data values based on historical data and parameters. Energy saving is achieved in the sense that only those sensor nodes whose sensed data values considerably deviate from the predicted values are allowed to transmit[49].
- **Low-Power Components**: Using low-power components like sensors, microcontrollers, and radio transceivers can save energy.
- **using energy-efficient communication protocols,** such as Zigbee, can help to minimize the energy consumption of IoT devices and WSN sensors. These protocols use low-power radio frequency (RF) transmissions that require less energy than traditional wireless protocols, such as Wi-Fi or Bluetooth.
- **routing** is a method for route between source and destination nodes. Key objective of routing protocols is to determine the shortest path that bring efficient energy consumption, less delay, the best service in order to expand the network lifetime via saving the sensors node alive more times as possible. [52]

II.7 Security and Privacy

Security and privacy challenges in wireless sensor networks and IoT are:

II.7.1 Data Confidentiality

To guarantee that confidential data gathered by sensors is never disclosed to unwanted listeners, it is necessary to maintain its secrecy. Under no circumstances should unauthorized parties have access to the gathered and transmitted data. To accomplish this, data encryption

with an exclusive key is implemented during the data gathering process, which is only known to the intended recipients and receivers. [53]

II.7.2 Source Authentication

the ability of ensuring the reliability of the collected and transmitted sensing data through the WSN's by verifying its source and data origin. Therefore, the communication turns out to be genuine since the malicious node cannot pretence in place of atrusted node. Consequently, Source Authentication is very important for decisions making and exchanging the control information of the WSN's.

II.7.3 Data Integrity

ensuring and confirming that the collected and transmitted data within the WSN's have never been tampered, altered, or corrupted by adversaries, malicious intermediate nodes, or even by accident due to the harsh communication environment of WSN's.

II.7.4 Availability

it ensures that the desired WSN's is available for the communication services and each node can use the network resources even with the attendance of denial-of-service attacks. Since the WSN's is collecting and transmitting the sensing data in charge of the communication services duty, the availability of WSN's is very essential for the survivability of IoT services and applications.

The CIA Triad of confidentiality, integrity and availability is considered the core underpinning of information security. Every security control and every security vulnerability can be viewed in light of one or more of these key concepts. For a security program to be considered comprehensive and complete, it must adequately address the entire CIA Triad. The figure II.12 shows the CIA triad [54].



Figure II.12: the CIA third .

To ensure confidentiality, integrity, and availability of data, various measures can be taken, including:

- **Access control**: This involves limiting access to data to only authorized personnel by implementing user authentication and authorization protocols.
- **Encryption:** Sensitive data can be encrypted to protect against unauthorized access. This involves converting plain text to coded data that can only be deciphered with a decryption key.
- **Backups and disaster recovery**: Regular backups can ensure that data is always available in case of loss, damage, or corruption. Disaster recovery plans can also be put in place to ensure that the data is promptly restored in case of a disaster.
- **Monitoring and logging**: Continuous monitoring of data activity can help detect any unauthorized access or activity. Logging can also track all actions taken on the data.
- **Physical security**: Access to data centres and servers can be protected using physical security measures such as closed-circuit cameras, biometric readers, and guards.
- **Regular updates and patches**: Regular updates to security software and systems can ensure that vulnerabilities are patched and the latest security protocols are in place.

II.8 Conclusion

In conclusion, the convergence of WSNs and IoT has brought about transformative changes in various sectors, including healthcare, agriculture, transportation, manufacturing, and smart cities, enabling us to collect vast amounts of data, improve decision-making, and enhance efficiency. By leveraging the power of connected devices, real-time monitoring, and intelligent systems, we can create smarter, more sustainable, and safer environments. However, it is crucial

to address the challenges of security, privacy, scalability, and standardization to fully unlock the potential of WSNs and IoT and ensure their responsible and effective implementation.

Chapter III:

Design and implementation of the system, tests and results.

III.1 Introduction

In this chapter, we will present the overarching concept of the project. Firstly, we will provide clear definitions for the various components employed, namely the ESP32 and the MPU6050. Next, we will delve into the programming tools utilized, including the Arduino IDE, Python, and various libraries. Additionally, we will elucidate the sequential steps undertaken to accomplish our objectives, encompassing dataset creation, data pre-processing, the architecture of the employed CNN model, model training, obtained results, and ultimately, the resolution of challenges encountered while implementing the same model on the embedded ESP32 platform.

III.2 Project conception

The project aims to develop a comprehensive system for monitoring the daily living activities of elderly individuals using Artificial intelligence technology. By leveraging IoT devices, sensors, and data analytics, the system will provide valuable insights into the well-being and safety of elderly residents within their homes. This project intends to enhance the quality of life for the elderly population while providing peace of mind to their caregivers and family members.

We've gone through three different phases of creating this project:

1. The first one is Data Collection; this part allows for us to create a dataset from different movements and position of an elderly such as walking, sitting, stand up, Lying down, and other positions. by using two sensors of movement (MPU6050), this type of sensors is used as accelerometer and gyroscope.
    a. The first sensor, with the processing unit, is used to detect movements of the upper part of the person's body, this sensor is fixed on the belt.
    b. The second sensor is used to detect movements on the lower part of the person's body, this sensor is fixed on a belt on the thigh as shown on figure III.1.

This system belongs to wearable devices categories. And we use it during dataset creation phase and as our project final wearable system.

2. The second phase involves dataset creation. The database being used is currently under construction and is affiliated with the Laboratory of Advanced Electronic Systems (LSEA) at the University of Medea. This work is part of an integrated project in the field of elderly care. We have made some contributions to this dataset using the system described earlier. In the subsequent sections of this chapter, we will provide a detailed description of this dataset.

3. The third phase involves data processing and analysis. In this stage, machine learning algorithms are utilized, specifically the CNN model (1D), for the classification of activities.
4. The fourth and final phase involves the implementation of real-time monitoring of activities and the secure transmission of data to a central server via Wi-Fi connectivity.

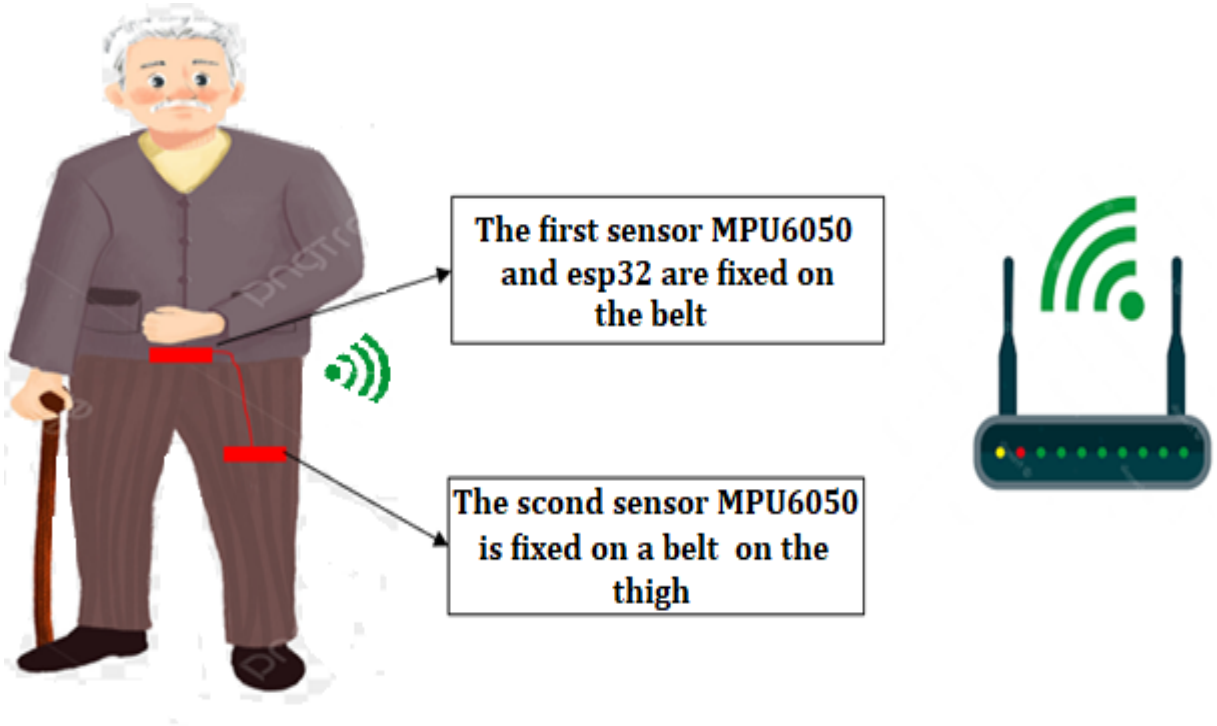


Figure III.1: synoptic diagram of system

III.3 Description of the components used.

In order to achieve the construction of our project we have used the ESP32 microcontroller because it equipped with built-in Wi-Fi capabilities, allowing it to connect to wireless networks and enable communication over the internet.

The ESP32 is a system-on-a-chip (SoC) that combines the features of both a microprocessor and a microcontroller. It is commonly referred to as a microcontroller due to its integrated microcontroller unit (MCU) along with other components such as Wi-Fi and Bluetooth modules. However, the ESP32's MCU is based on the Xtensa LX6 microprocessor architecture, which allows it to perform tasks traditionally associated with microprocessors. So, the ESP32 can be considered as a hybrid device that combines the capabilities of both microprocessors and microcontrollers.

Additionally, we have incorporated two MPU6050 sensors due to their remarkable capabilities in motion sensing and orientation tracking. These sensors are capable of providing precise measurements for acceleration, angular velocity, and temperature. In the following sections, we will elaborate on these components in greater detail.

III.3 .1 Esp32

ESP32 is a low-cost, low-power system on a chip (SoC) microcontroller with integrated Wi-Fi and Bluetooth capabilities. It is produced by Espressif Systems. The ESP32 can be programmed using various languages, including the Arduino IDE and Micro Python, and other programming languages. This microcontroller and has become popular for its ease of use.It can be utilized in wide range of applications, such as home automation, industrial automation, smart appliances, and more. It comes with a

- Micro USB interface that allows for code uploading and power application by connecting it to the computer.
- USB to UART converter that can establish communication between the computer and the board.
- RESET (which may also be labeled EN) that functions to initiate a restart of the board.
- BOOT button that can be utilized to put the board into flashing mode.
- Blue LED that is directly connected to GPIO 2.
- Red LED that illuminates when the board is given power. (Figure III.2).

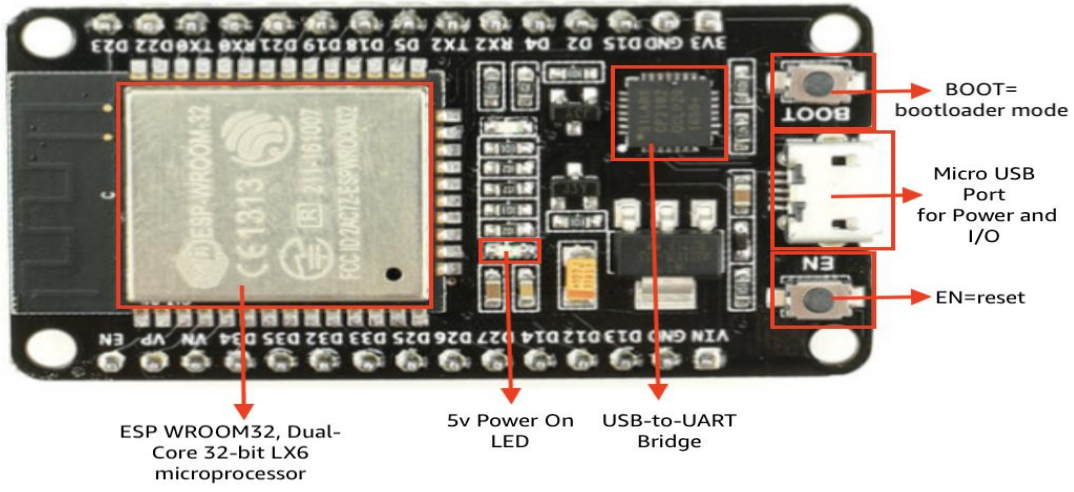


Figure III.2 ESP32 board.

The main characteristics of the ESP32 are:

- CPU: The ESP32 features a dual-core Tensilica LX6 microprocessor
- Memory: 520KB SRAM, 448 KB of ROM, which is expandable up to 16MB supports multiple peripherals such as SPI, I2C, and UART.
- has integrated ADCs, DACs,
- versatile microcontroller for a range of projects.
- operates within a range 2.2V to 3.6V
- wireless connectivity
  - Wi-Fi: supports both 2.4 GHz and 5 GHz frequency bands.
    connect the ESP32 to local networks or the internet
  - Bluetooth: supporting both classic Bluetooth and Bluetooth Low Energy (BLE) protocols. It enables the ESP32 to communicate with other devices that support Bluetooth, such as smartphones, tablets, or other IoT devices.
- includes power management features, such as sleep modes and power-saving techniques, which help optimize power consumption for battery-powered applications [55].

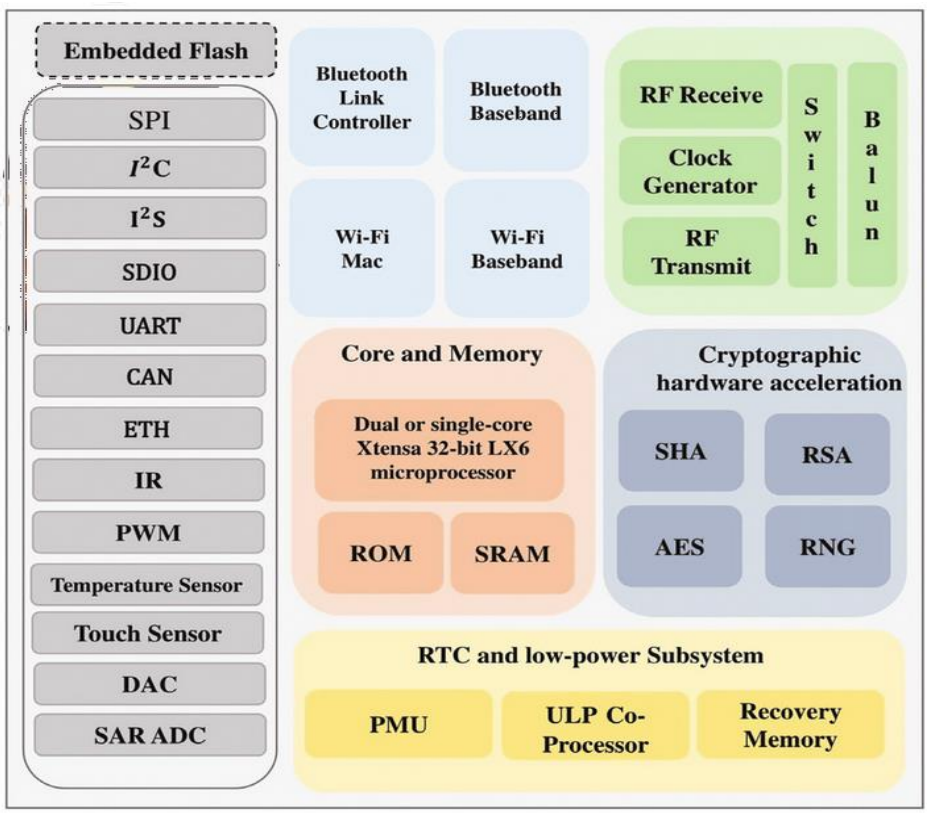


Figure III.3: ESP32 function block diagram.

III.3.2 MPU 6050:

The MPU6050 is a popular sensor module that combines a 3-axis gyroscope and a 3-axis accelerometer onto a single chip. This sensor measures the orientation and motion of an object in real-time. The accelerometer measures the gravitational acceleration, and the gyroscope measures the rotational velocity. the module also measures the temperature. The MPU6050 communicates with the microcontroller through I2C protocol and provides digital output in the form of acceleration and angular velocity. It is a popular choice for a variety of purposes, including motion tracking, gaming, drones, and robotics.



Figure III.4: The MPU6050 module.

a) The Gyroscope

The gyroscope measures the rotational velocity of an object, denoted in rad/s. It tracks the change in angular position over time along each of the X, Y, and Z-axis, also known as roll, pitch, and yaw. This ability to measure the changes in position enables us to ascertain the orientation of an object. The figure III.5 illustrate the Gyroscope axis (roll, pitch and yaw).

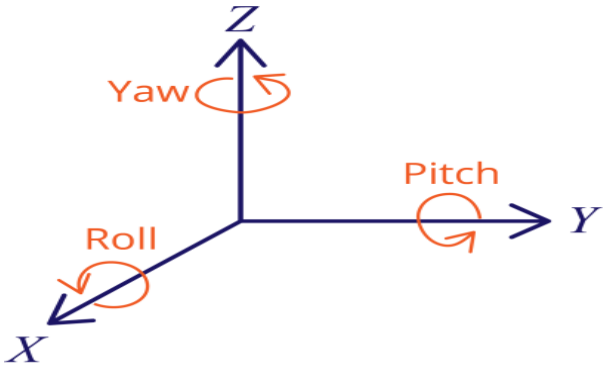


Figure III.5: The Gyroscope axis (roll, pitch and yaw).

b) The accelerometer:

The accelerometer measures the acceleration of an object, which is the rate at which the object's velocity changes. The accelerometer is capable of detecting both static forces such as gravity ($9.8m/s^2$) and dynamic forces such as movements or vibrations. The MPU-6050 measures acceleration over the X, Y, and Z-axis. where the object being measured is static, the acceleration over the Z-axis is equal to the force of gravity, and it should register zero on the X and Y-axis.

c) Block diagram of the MPU6050 circuit

The components and blocks work together to enable the MPU6050 to measure acceleration and angular velocity, process the sensor data, and provide it to the external microcontroller or system for further analysis and control. the figureIII.6. gives a detailed description of the block diagram of the MPU6050 circuit:

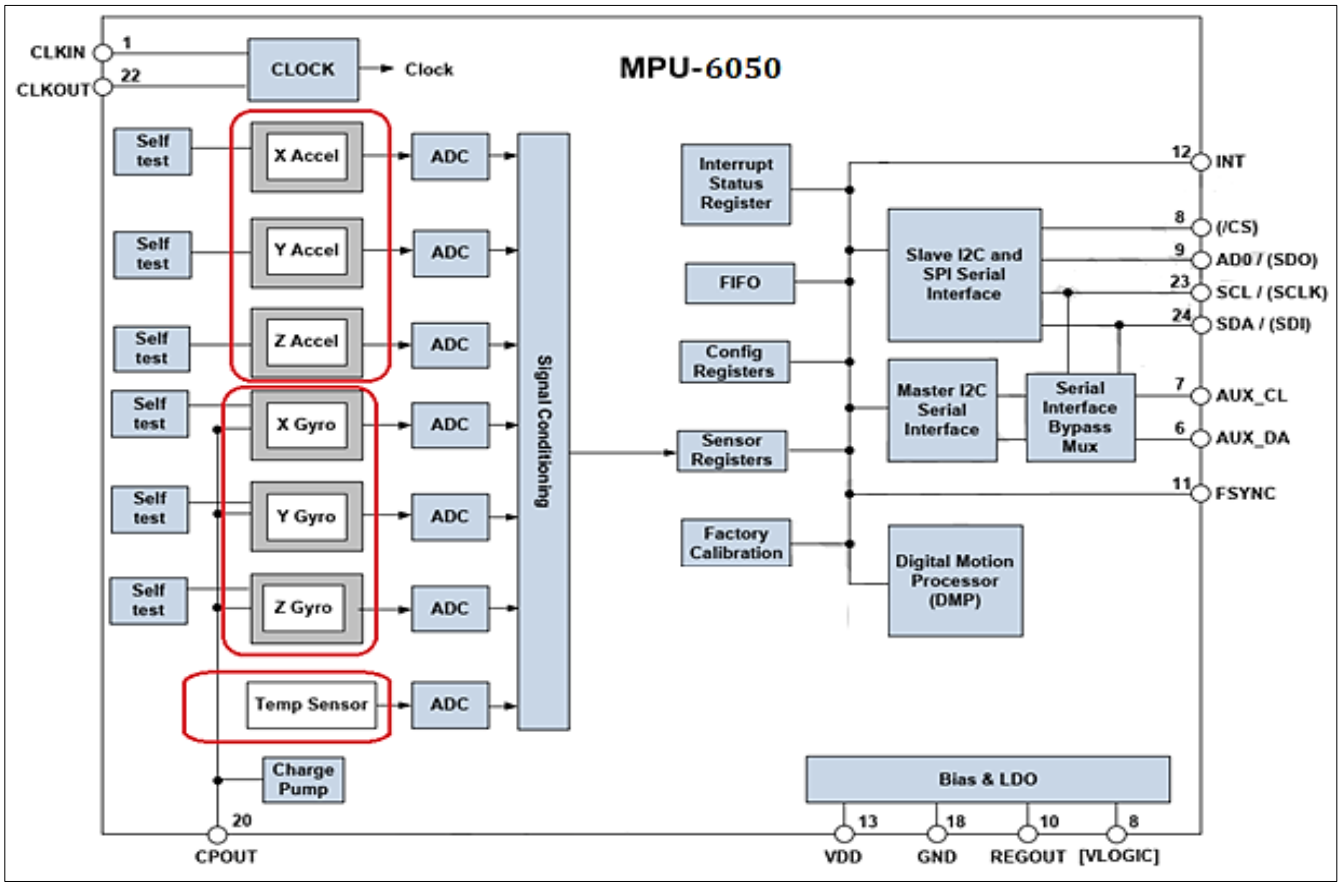


Figure III.6: The block diagram of the MPU6050 circuit

III.4 System architecture

Our system architecture contained a single ESP32 to acquire data from the two MPU6050 sensors through the I2C interface. It sends appropriate commands and requests to the sensors to obtain the desired data, such as acceleration and angular velocity.

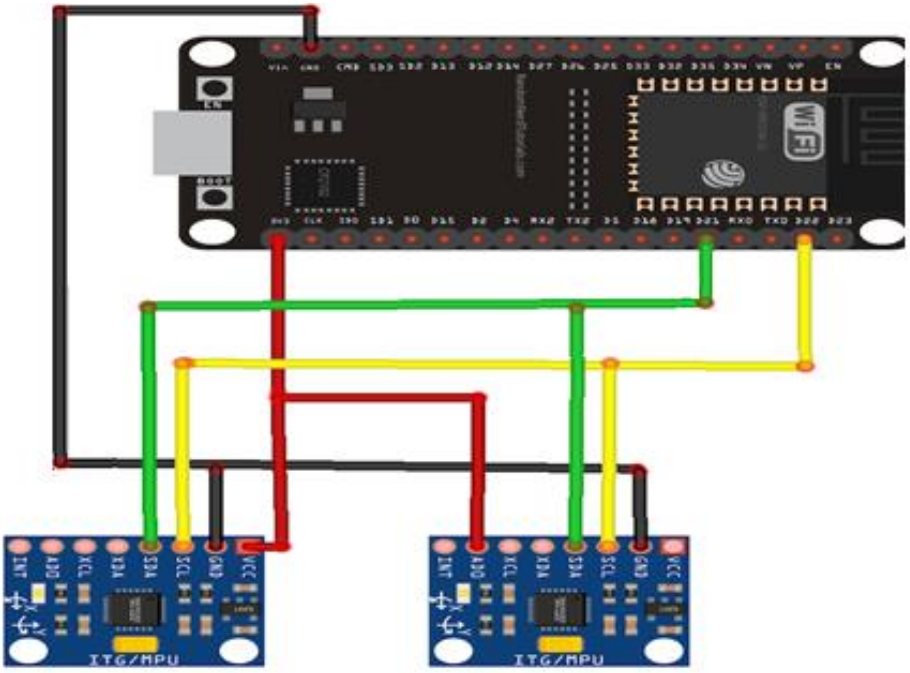


Figure III.7: The System architecture

VCC: to power the capture at 3.3V or 5 V, we used 3.3v.

SDA: the SDA branch for I2C protocol communication (GPIO21).

SCL: the SCL branch for I2C protocol communication (GPIO22).

GND: common mass

AD0: This branch used to select the I2C address, we connect the second capture to VCC for change the address of this capture.

Table III.1: Connections between the components.

The connection colour	ESP32	MPU6050_ (1)	MPU6050_ (2)
Rad	VCC	VCC=3.3	AD0
Green	SDA	SDA	SDA

Yellow	SCL	SCL	SCL
Black	GND	GND	GND

III.4.2 The communication protocol between ESP32 and MPU6050

I2C (Inter-Integrated Circuit) is a widely used serial communication protocol that allows multiple devices to communicate with each other using a two-wire interface. It was developed by Philips (now NXP Semiconductors) and has become a standard protocol for connecting various integrated circuits on a common bus.

The I2C protocol uses two lines for communication:

1. SDA (Serial Data): This bidirectional line is used for transmitting and receiving data between devices. It carries the actuel information being communicated.
2. SCL (Serial Clock): This line is used to synchronize the data transfer between devices. The clock signal is generated by the device acting as the master and controls the timing of data transmission [56].

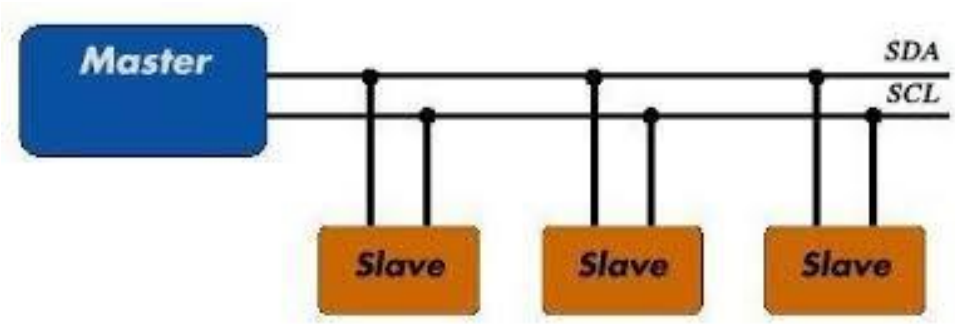


Figure III.8: The I2C protocol.

With the ESP32, we can easily interface with I2C by using some necessary libraries such as “wire.h” in Arduino ide

I2C offers a simple and efficient way to connect multiple devices using only two wires, making it a popular choice in embedded systems and small-scale communication scenarios.

III.5 The software used

To accomplished this work, verities of software are used:

III.5 .1 The Arduino IDE

Arduino is an open-source platform That provides a user-friendly interface to write, compile, and upload code onto the Arduino board. The IDE comprises

- a text editor.
- a compiler.
- a bootloader.

This comprises permits users to design and upload programs onto the Arduino board with ease.

- Arduino IDE in enabling us to visualize and analyze the various sensor readings and signals in real-time.
- Arduino IDE offers a user-friendly interface and a robust set of tools, making it an ideal choice for monitoring and analyzing signals in our project.
- we leveraged the capabilities of Arduino IDE to develop a comprehensive and efficient signal monitoring system for our project.
- by using Arduino IDE, we were able to acquire, process, and interpret data from different sensors, enabling us to gain valuable insights into the daily activities of the elderly individuals.
- Arduino IDE served as a crucial tool in our project, allowing us to validate and refine our monitoring algorithms and ensure the accuracy and reliability of the system.

a) The utilisation and configuration of Arduino ide

The programming software for the code ESP32 (C++-like language). Once the program is typed or modified on the keyboard, it will be transferred and stored in the card through the USB link. The USB cable powered both the card and also carries the information, this program called IDE Arduino. The Arduino software is an open source and free development environment (IDE) downloadable from Arduino's official website. The Arduino IDE enables:

- To edit a program: sketches (sketch).
- To compile this program in the "machine language" of the Arduino,

- To upload the program into the memory of the Arduino,
- Communicate with the ESP32 card through the terminal.

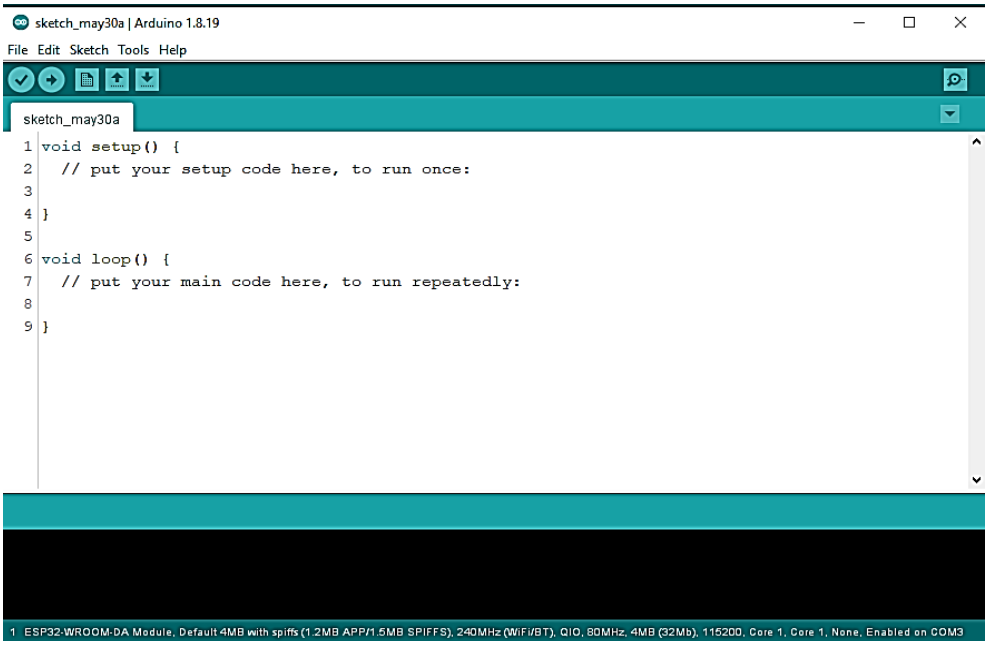


Figure III.9: Arduino IDE environment

b) The installation of ESP32 carte in Arduino IDE

To install the esp32 carte in Arduino IDE, we need to add support for the ESP32 boards. Follow the next steps:

1: we will go to **File > Preferences**.

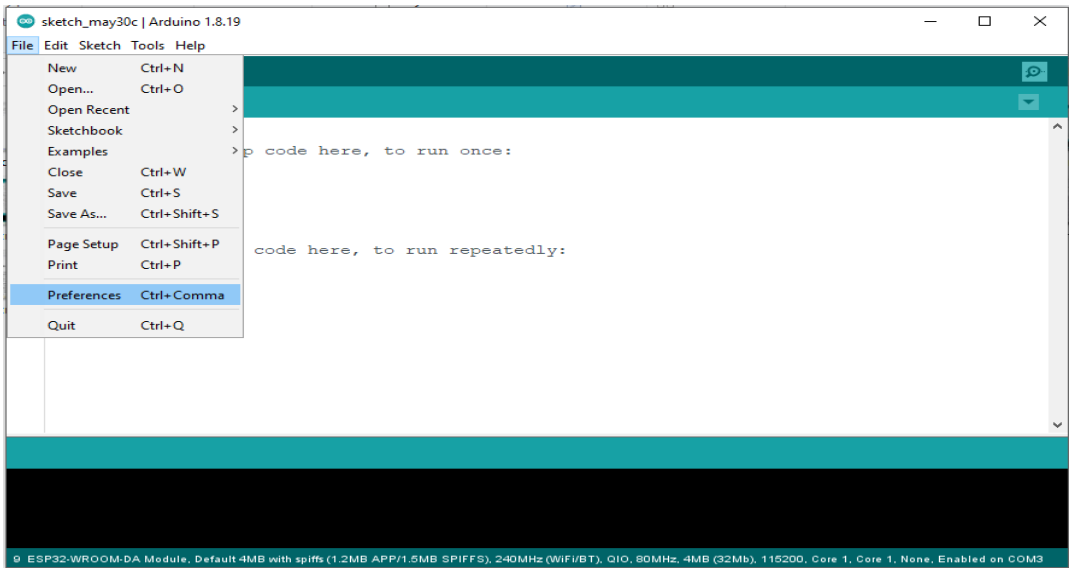


Figure III.10: the installation of ESP32 in the IDE.

2: we will add support for ESP32 by enter the following into the "*Additional Board Manager URLs*"field. https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json.

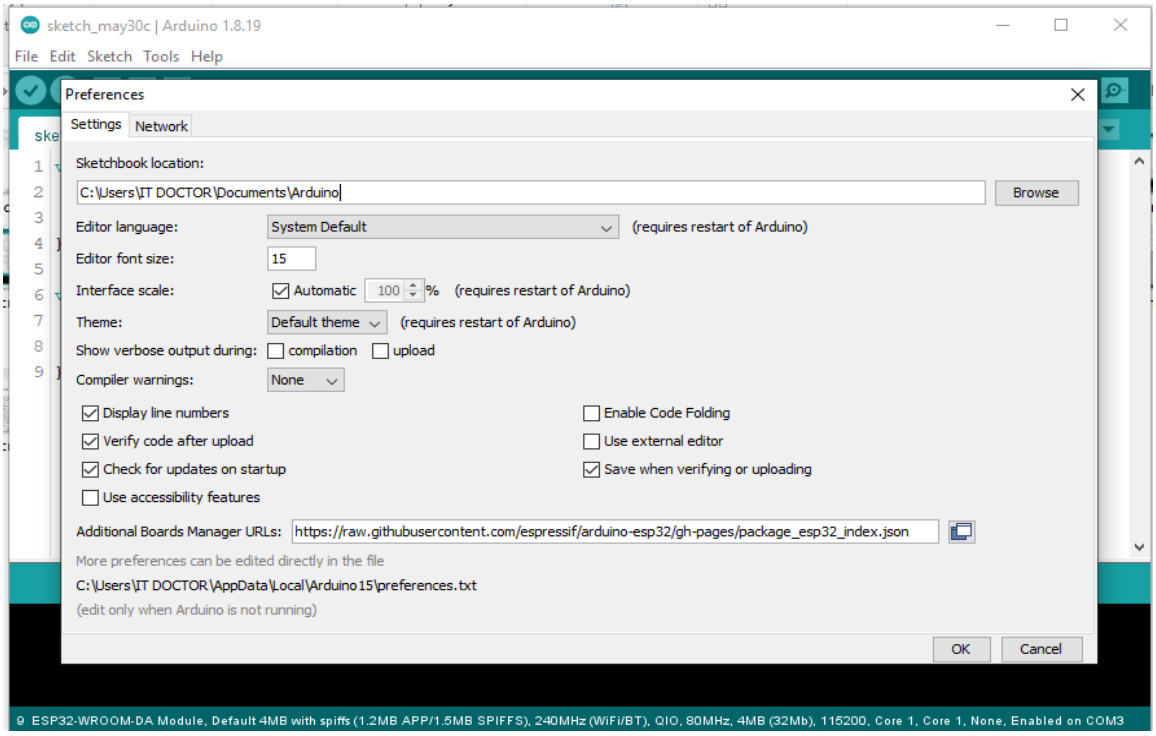


Figure III.11: Installation of ESP32 in the IDE.

3: We Open the **Boards Manager**. Then we go to **Tools > Board >Boards Manager.**

4: We Search for **ESP32** and we install the "**ESP32 by Espressif Systems**".

5: We go to **Tools > Board** and check we have used ESP32 DEVIKIT V1 and select the port COM 3.

c) Installation of the library of MPU6050

For install the library of MPU6050, we need to open our Arduino IDE and we go to **Sketch > Include Library > Manage Libraries.** To install the library, simply search for "adafruit mpu6050" on the search box and proceed with the installation process.

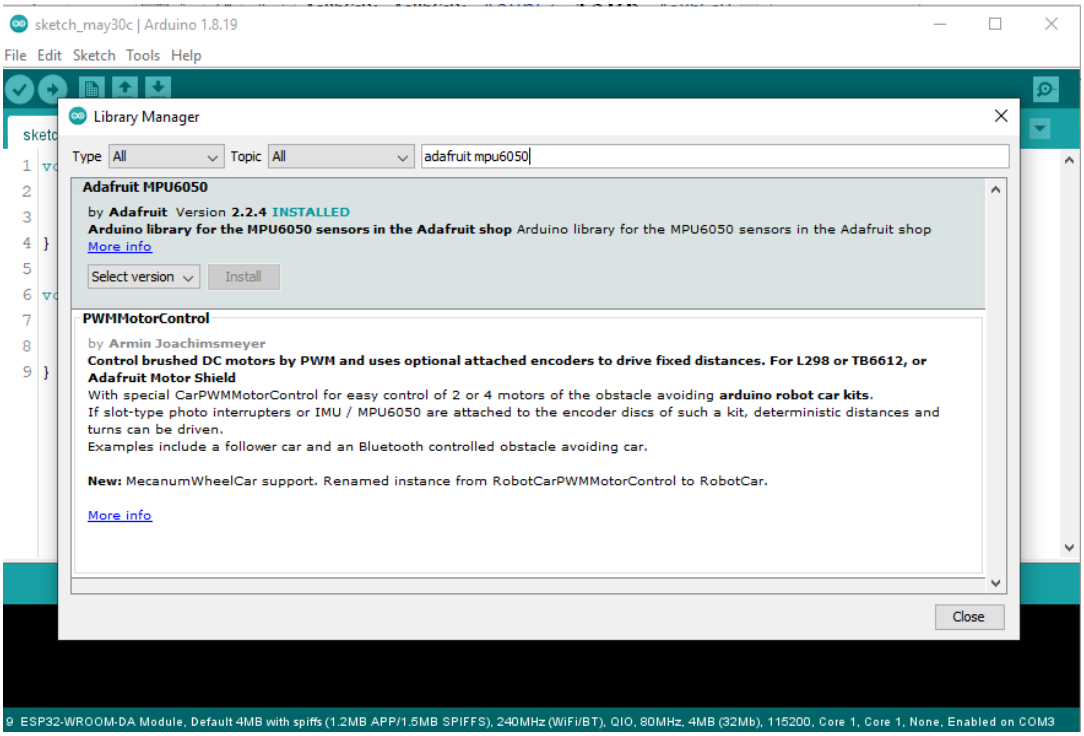


Figure III.12: installation the library of the MPU6050.

Then, we search for the library **"Adafruit Unified Sensor**" and install it.

Finally, we search for "**Adafruit Bus IO**" and install it.

III.5 .2 Python

Python is a high-level programming language that is widely used in various domains that include scientific computing, machine learning, data analysis, web development, and more.

Guido van Rossum was created This language in the late 1980s and is named after the comedy group Monty Python.

the syntax of Python is designed to be

- easy to read and write
- making it an ideal language for beginners
- a large community of developers and users

the developers create and share the libraries, frameworks, and tools, making development faster and easier.

Python is classified as an interpreted language, this mean that the code is executed line-by-line, rather than being compiled into machine code.

We have used python to save the change that give for us by IDE form a csv files.

III.5 .3 Libraries used

a) 1.TensorFlow

TensorFlow is a software library that is available for open-source use .it also supports dataflow and differentiable programming. Originally developed by the Google Brain team. Computational graphs are used to represent complex mathematical computations, with nodes representing operations and edges representing the relationships between them. TensorFlow can support several programming languages, including Python, C++, and Java and others TensorFlow is

- applicable to a wide variety of tasks
- a popular tool for creating deep learning models and other machine learning applications
- versatile

Its scalability and flexibility make it suitable for a wide range of industries, from natural language processing to computer vision.

b) Keras

Keras is an open-source deep learning framework written in Python. It provides a high-level API for building and training deep neural networks, making it easy to quickly prototype and experiment with different models. Keras is designed to be

- user-friendly,
- modular,
- extensible.

This means that developers can create intricate deep learning models with just a few lines of code. It also has a large community of users and contributors, which means that there are lots of resources and tutorials available online .it can run on both CPU and GPU, making it suitable for a wide range of applications.

c) sklearn

short for Scikit-learn, is an open-source Python library used for machine learning and data analysis. the data scientists, researchers, and machine learning practitioners used sklearn because it simple, easy, and powerful capabilities. It includes a range of functions for:

- classification,
- regression,
- clustering,
- dimensionality reduction,
- model selection,
- and data pre-processing.

Some of the most commonly used algorithms that are available in sklearn include Linear Regression, Logistic Regression, Random Forest, and K-Means. Sklearn also provides tools for model evaluation and selection, cross-validation, and hyperparameter tuning.

d) NumPy

NumPy is a Python library that is mainly used for working with arrays and matrices of numerical data. It provides various mathematical functions, operations and transformations for these arrays, making it a powerful tool for scientific computing, data analysis and machine learning. NumPy is

- Efficient and fast array processing functions, especially for large arrays
- Support for multi-dimensional arrays and matrices
- A wide range of operations for manipulating arrays and matrices, including indexing, slicing, sorting and filtering

NumPy integrate with other Python libraries for data analysis and visualization, such as Matplotlib and Pandas.

e) Matplotlib

Matplotlib is the popular library in data visualization for the Python programming language, which is provides a wide variety of graphs, charts, and plots, ranging from basic scatter plots to complex 3D visualizations. Matplotlib is

- easy to use

- create high-quality visualizations from their data
- core functionality.

Matplotlib also provides a range of tools for customizing plots and visualization options.

III.6 The dataset

III.6.1 Dataset preparation

The dataset used for this study was collected at the Laboratory of Advanced Electronic Systems (LSEA). To gather the required data, the wearable system depicted previously is used. Its sensors were designed to capture changes in acceleration and angular velocity along the X, Y, and Z axes in various positions, including walking, sitting, standing up, lying down, going up and down the stairs, as well as other positions.

The data collection process involved using the ESP32 connected to a computer via a Wi-Fi connection. During the data collection phase, the received sensors values were printed in the Serial Monitor, allowing for real-time monitoring and verification. Subsequently, these values were stored in CSV files using Python code for further analysis.

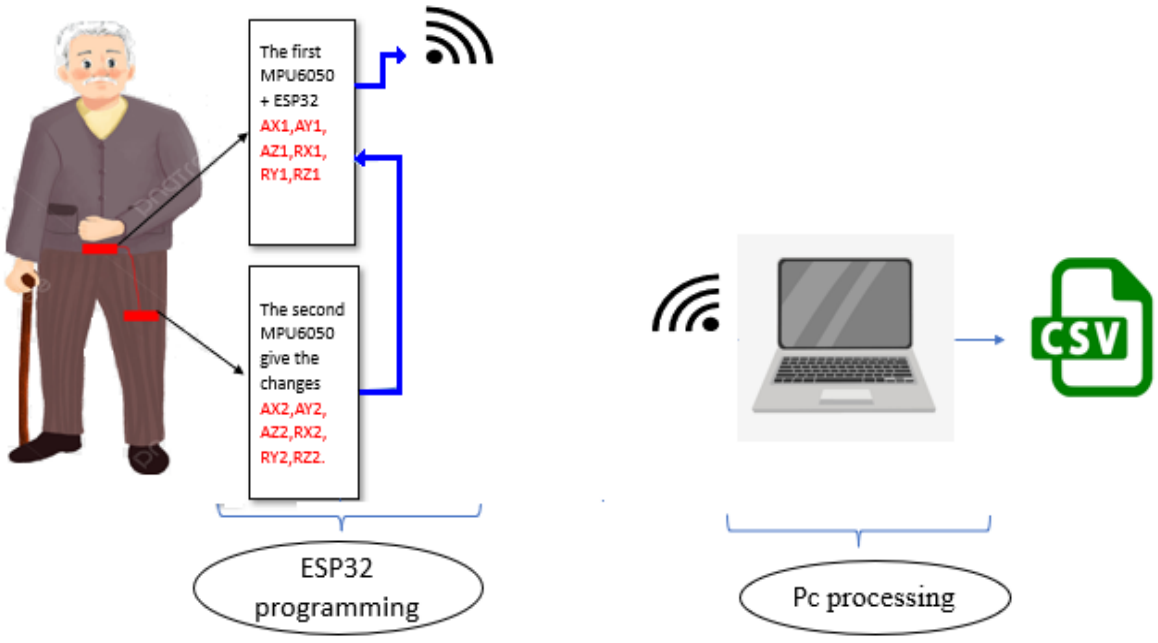


Figure III.13: diagram of data collection

III.6.2 data description

The dataset consists of 10 CSV folder, with each file representing a specific position of an elderly individual, as indicated in Table III.2.

Table III.2: the different position

Position name	Position label
Walking	0
Setting	1
Stand up	2
Lying down	3
Up the stairs	4
Going down the stairs	5
Transition from sitting to standing	6
Transition from standing to sitting	7
Going to sleep	8
Stand up from sleep	9

III.6.3. CSVs file Description

The dataset comprises multiple folders, and each folder contains many .CSV files. Each CSV file represents a particular position.

Each file contains 12 variants, specifically AX1, AY1, AZ1, RX1, RY1, RZ1, AX2, AY2, AZ2, RX2, RY2, and RZ2. These variants represent measurements of acceleration and angular velocity along the X, Y, and Z axes recorded by two sensors. Each file encompasses a 10-second duration, with data captured at a frequency of one acquisition every 50ms. This results in a total of 200 acquisitions or iterations within the file. Consequently, this duration enables a time-series representation of the motion data captured for the specified position.

P1MC0.csv

	A	B	C	D	E	F	G	H	I	J	K	L
1	AX1	AY1	AZ1	RX1	RY1	RZ1	AX2	AY2	AZ2	RX2	RY2	RZ2
2	-2.25	-10.74	2.41	-0.05	-0.59	-0.06	0.84	-10.53	2.34	-0.98	-0.77	0.5
3	1.64	-10.19	1.13	-0.25	-0.19	0.26	0.95	-11.06	0.08	-0.29	-0.75	0.26
4	0.72	-14.26	3.5	0.04	-0.26	0.43	-0.31	-14.61	-2.42	0.52	-0.29	-0.08
5	0.25	-8.21	4.17	0.59	-0.17	0.39	-4.83	-12.65	-2.93	3.63	-0.63	-1.87
6	-0.39	-9.4	4.51	0.19	-0.05	-0.09	6.57	-15.6	8.27	3.01	-3.07	-0.5
7	0.17	-9.75	3.31	0.71	0.29	-0.62	4.15	-11.45	4.31	2.38	2.55	-0.2
8	-1.2	-7.6	2.81	0.6	0.62	-0.62	0.43	-9.48	3.57	2.09	1.55	-0.45
9	-1.79	-6.15	2.08	0.37	0.61	-0.47	0.53	-7.35	4.6	1.52	1.59	-0.11
	....	....	....	....	....	....	....	....	....	....	....	....
	....	....	....	....	....	....	....	....	....	....	....	....
200	-1.96	-7.83	0.79	-0.15	0.39	0.01	-0.07	-7.01	2.75	-0.31	0.55	0.42

Figure III.14: An example of .csv file.

The figure III.15 shows the acceleration and rotational of the upper and the lower side of the elderly.

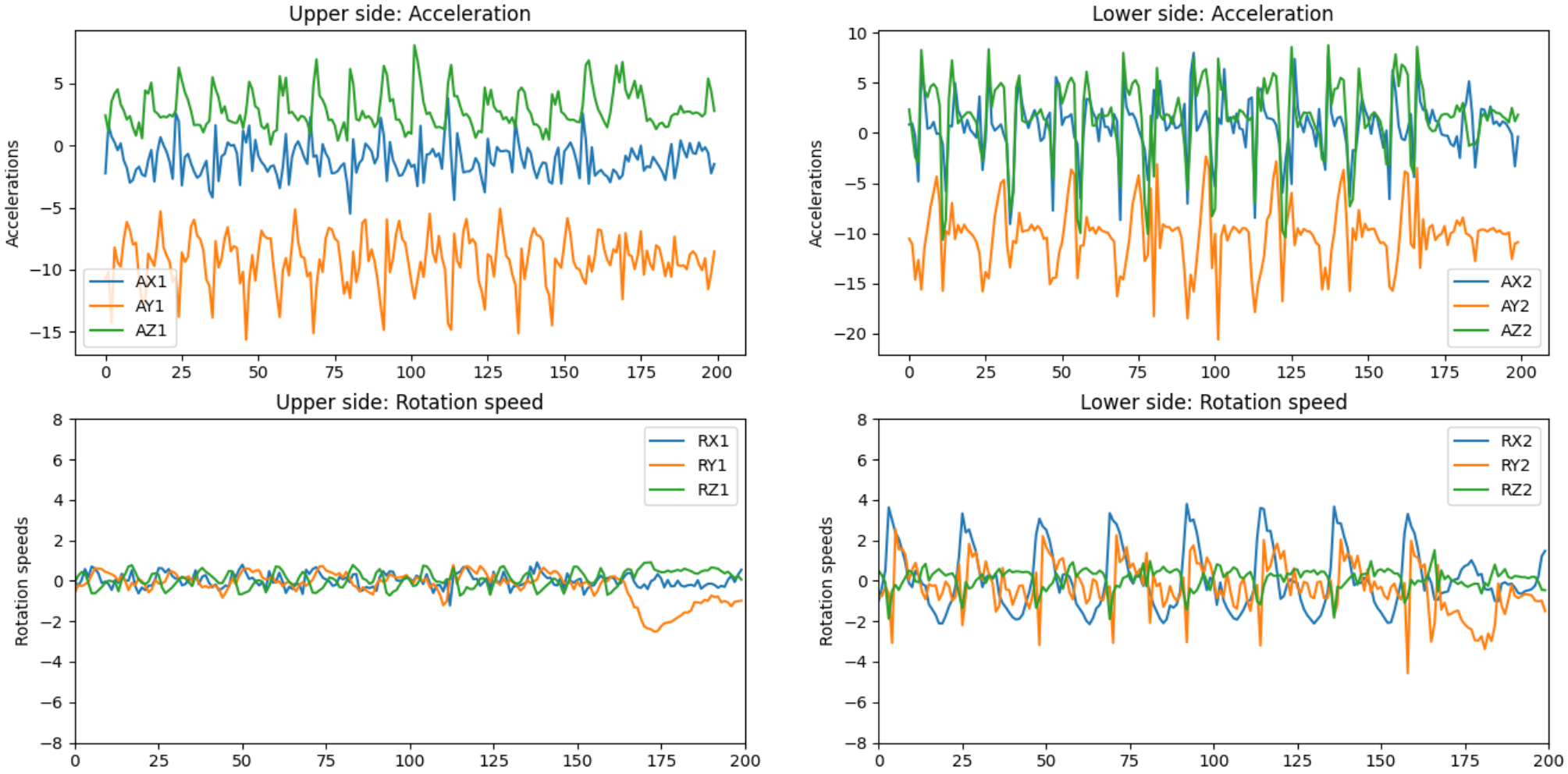


Figure III.15: Accelerations and rotations for upper and lower part of body during walking.

III.7 The CNN model

We have created several CNN architectures (model) in order to choose the one that offers the best characteristics (accuracy, number of parameters, etc.). Figure III.16 represents the different steps to follow to train the CNN network.

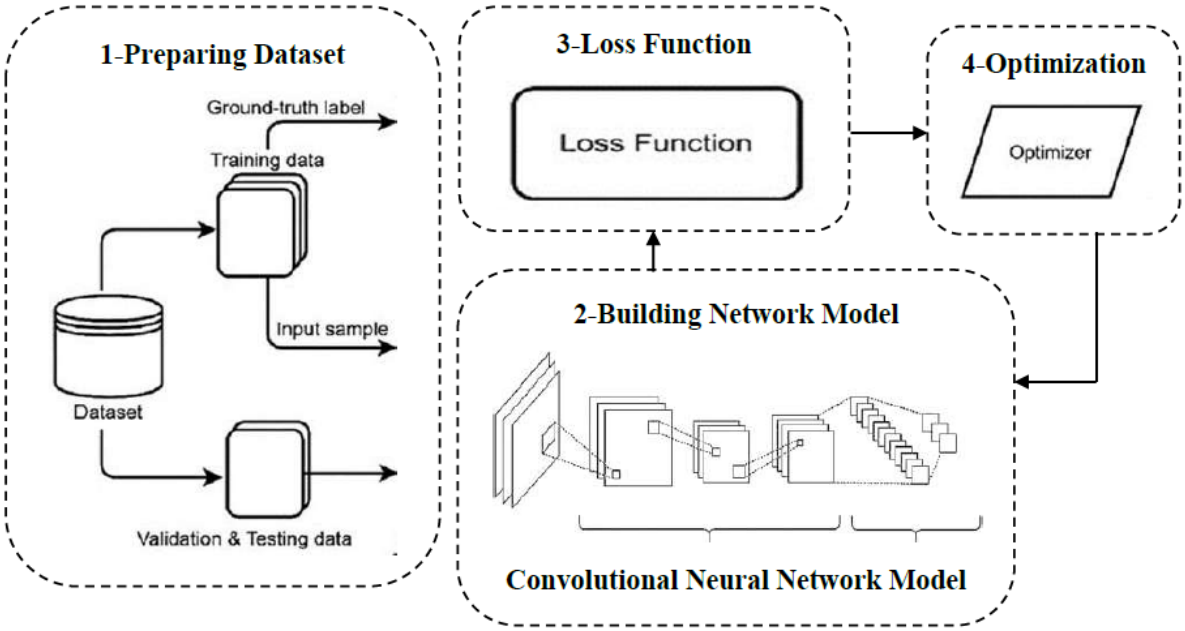


Figure III.16: different steps to follow to train the CNN network.

III.7.1 Import packages.

First step in Python program is to import the necessary libraries:

- "Pandas" for data manipulation.
- "Numpy" for numerical calculations.
- "Matplotlib" for visualization of results .
- "Tensorflow" for machine learning.
- "Keras" for building and training the models.

```
import tensorflow
from tensorflow import keras
import pandas
import numpy as np
#import seaborn as sns
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Flatten, MaxPooling2D, BatchNormalization, AveragePooling2D, Conv1D
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
from keras.callbacks import ModelCheckpoint
#import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from keras.models import load_model
from keras.models import model_from_json
from keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
from keras.layers import Dropout
from keras.utils.np_utils import to_categorical
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from keras.callbacks import ModelCheckpoint
```

Figure III. 17: Importing the necessary libraries

III.7.2 Pre-processing of input data from CSV files in each directory

We have created a code that can iterates through the directories and files in the given path, reads the CSV files using Pandas, and appends the data to the "**data**" list. It also assigns a target label to each set of data based on the directory it belongs to. Finally, it prints the length and shapes of the data and target arrays

```
path = '/content/drive/MyDrive/DLA_Dataset/'
mylist= os.listdir(path)
mylist.sort()
print(mylist)
N_Classes=len(mylist)
data=[]
N_C=[]
T=0
tr=np.array([])
for x in mylist:
  mypiclist=os.listdir(path+"/"+str(x))
  for y in mypiclist:
    #print("y=",y)
    curimg=pandas.read_csv(path+"/"+str(x)+"/"+y)
    curimg=curimg.values
    a=np.array([])
    for i in range(0, 200, 4):
          a=np.append(a,curimg[i,:])
    data.append(a)
    tr=np.append(tr,T)
    a=np.array([])
```

Figure III.18: processing the CSV files in each directory

In this case, the Input data is a Numpy arrays of (1,2400) which mean that we have aligned the 200 row of 12 columns of each csv file in a single row of 2400 column. (2400= 600*12).

1) Decomposition of data

After we processing the CSV files, we have split the data into 80% for training and 20% will be allocated to the testing set.

This line splits the arrays x and y into training and testing sets. The training set is assigned to '**xtrain**' and '**ytrain**', while the testing set is assigned to **'xtest'** and **'ytest'**. (figure III.18.)

```
#train, val, ctr, cval = train_test_split(x, y, test_size=0.2)
xtrain, xtest, ytrain, ytest=train_test_split(x, y, test_size=0.2)
```

Figure III.19: Splitting the datasets into training and testing sets.

III.7.3 Architecture used

After numerous attempts, we finally succeeded in crafting an exceptional architecture that consistently delivers impressive outcomes.

The architecture used is compose from:

- Two layers of convolution.
- One layer of batchNormalisation.
- One Flatten layer
- 3 Dense layers
- One Dropout layer

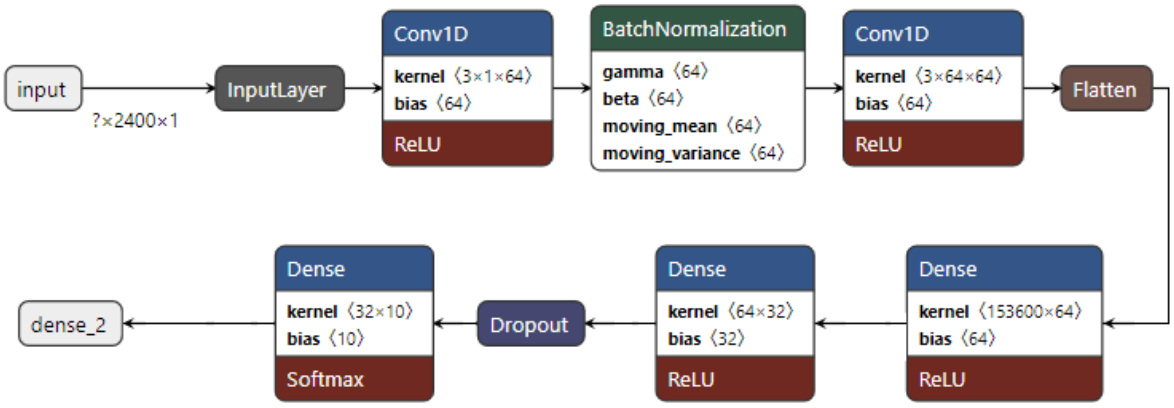


Figure III.20: The architecture of the CNN model

In the first time we provide a one-dimensional convolutional layer with 64 filters, a kernel size of 3, padding set to **'same'** (which adds padding to ensure the output has the same length as the input), and an activation function set to **'Relu'**. The input shape of this layer is (2400, 1), indicating that the input data has 2400-time steps and 1 feature. then the **BatchNormalization** normalizes the activations of the **Conv1D** layer by subtracting the batch mean and dividing by the batch standard deviation. It helps in stabilizing and speeding up the training process. we have used **a**nother one-dimensional convolutional layer with 64 filters, a kernel size of 3, padding set to 'same', and an activation function set to **'Relu'.** Then we flatten the output of the previous convolutional layer into a one-dimensional array by used the **'flatten'**, preparing it for input into a fully connected layer. The fully connected layer is with 32 units (neurons) and an activation function set to **'Relu'.** It takes the flattened output of the previous layer as input,another fully connected layer with 64 units and an activation function set to 'relu'. Also, we used a Dropout layer: This layer randomly sets a fraction of input units to 0 at each update during training, which helps prevent overfitting. In this case, 40% of the input units are set to 0. The final fully connected layer with 10 units and an activation function set to 'softmax'. This layer outputs probabilities for the 10 possible classes of the classification problem.

The model is compiled using categorical cross-entropy as the loss function, Adam optimizer, and accuracy as the evaluation metric. The **summary()** function displays a summary of the model's architecture, including the number of parameters in each layer and the total number of trainable parameters.

The next table show the two architectures.

Table III.3: A comparison between two architectures.

	Architecture 1	**Architecture 2**
	Conv1D (64,3) Padding	Conv1D (64,2) Padding
	BatchNormalization	BatchNormalization
	Conv1D (64,3)	Conv1D (64,2)
	Flatten	Flatten
	Dense(64)	Dense(64)
	Dense(32)	Dense(32)

	Drapout (0.4)		Drapout (0.4)	
	Dense(10)		Dense(10)	
Training	Accuracy	75%	Accuracy	97%
	Loss	0.69	Loss	0.05
Validation	Accuracy	82%	Accuracy	99%
	Loss	0.5	Loss	0.02

As mentioned earlier, our journey involved exploring two different architectural approaches, both of which yielded commendable validation accuracy. However, we encountered a significant hurdle when attempting to implement our model on the ESP32. The ESP32's limited random access memory capacity proved insufficient to accommodate our intricate model. This limitation arose from the size of the input array (1*2400) of float numbers in the first layer of our model and the number of kernels used which is 64.

We embarked on a quest to find a viable solution. We devised a strategy to divide each file into four distinct sections, thereby creating a total of four input arrays from each file. As mentioned in the following figure.

By embracing this approach, we successfully reduced the size of each input from (1*2400) to (1* 600) while simultaneously increased the number of inputs in our dataset. In addition, we have changed the default type of the float numbers used by python from **float64** to **float32**. This adjustment allowed us to overcome the memory limitations and fully unleash the potential of our model.

```
tr=np.array([])
for x in mylist:
  mypiclist=os.listdir(path+"/"+str
  for y in mypiclist:
    #print("y=",y)
    curimg=pandas.read_csv(path+"/"
    curimg=curimg.values
    a=np.array([])
    for i in range(0, 200, 4):
         a=np.append(a,curimg[i,:])
    data.append(a)
    tr=np.append(tr,T)
    a=np.array([])

    for i in range(1, 200, 4):
         a=np.append(a,curimg[i,:])
    data.append(a)
    tr=np.append(tr,T)
    a=np.array([])
    for i in range(2, 200, 4):
        a=np.append(a,curimg[i,:])
    data.append(a)
    tr=np.append(tr,T)
    a=np.array([])
    for i in range(3, 200, 4):
         a=np.append(a,curimg[i,:])
    data.append(a)
    tr=np.append(tr,T)
    a=np.array([])
  T=T+1
print('len(data)= ',len(data))
tr = tr.reshape(tr.shape[0],1)
print('tr.shape =',tr.shape)

d=np.array(data)
```

Figure III.21: Splitting each .csv file into 4 inputs array of (1* 600).

In addition, a new simplified CNN model (in terms of kernel size) has been implemented as shown in the figure below:

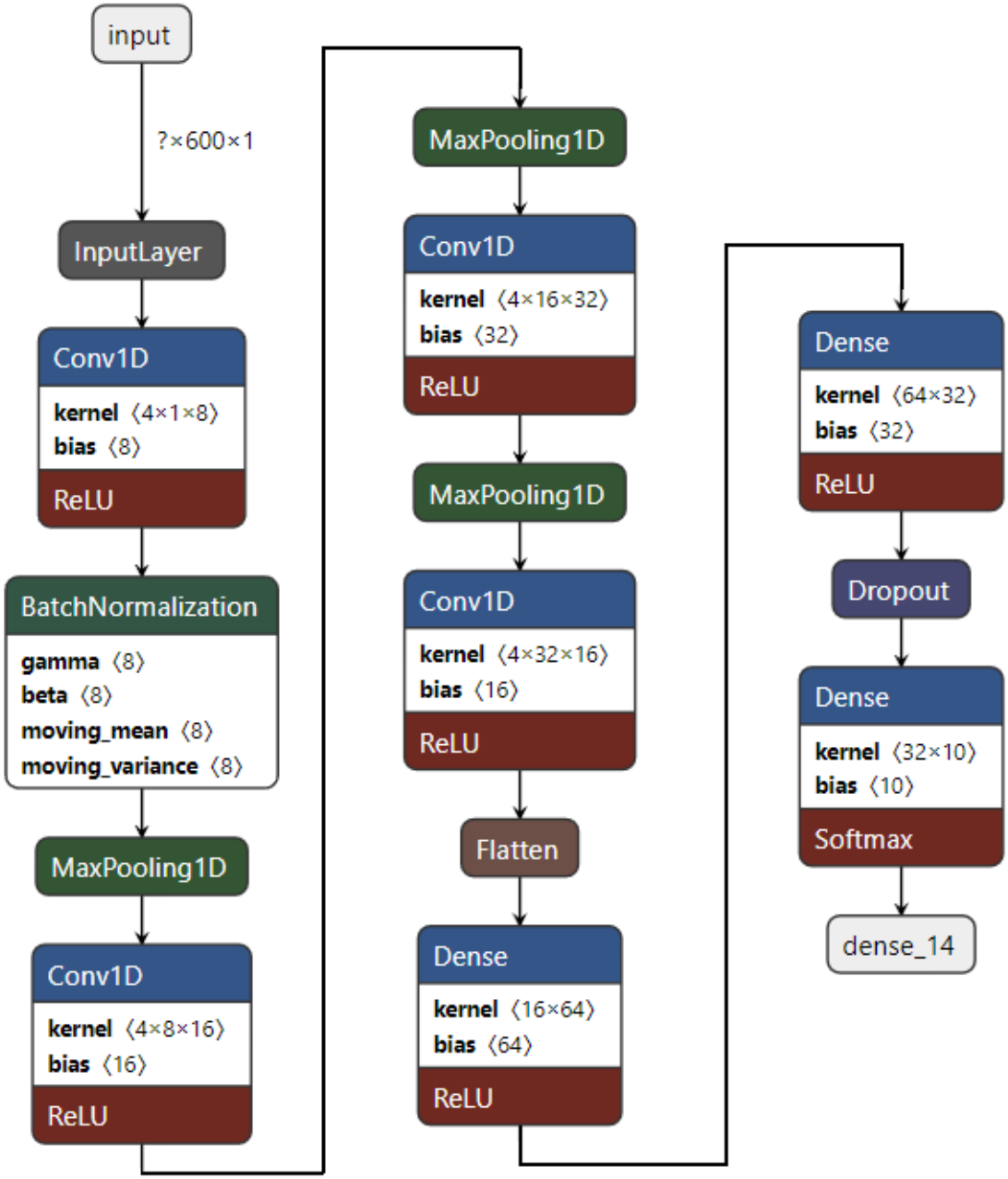


Figure III.22: The new implemented architecture.

III.7.4 Save the best model

By using the **'ModelCheckpoint callback'**, we can automatically save the weights of the best model during training, which can be useful for later evaluation or deployment.

```
# Define the ModelCheckpoint callback
checkpoint = ModelCheckpoint(checkpoint_filepath, monitor='val_accuracy', save_best_only=True, mode='max', verbose=1)
```

Figure III.23: Model Checkpoint creation.

By executing this code, we will visualize the training and validation loss as well as the training and validation accuracy over epochs, helping us to understand the model's performance during training, and save it with ".**h5**"

```
# Train the model with the callback
history=classifier.fit(xtrain, ytrain, validation_data=(xtest, ytest), batch_size=64, epochs=100, callbacks=[checkpoint])
classifier.load_weights(checkpoint_filepath)
classifier.save('/content/drive/MyDrive/model.h5')
```

Figure III.24: saving the model ".**h5**"

III.7.5 The obtained results

In order to show the results obtained by our CNN 1D model, we illustrated the different results in terms of

- Accuracy.
- loss.
- Confusion Matrix.

a) Accuracy and losses graph

By executing this code, we will visualize the training and validation loss as well as the training and validation accuracy over epochs, helping us to understand the model's performance during training.

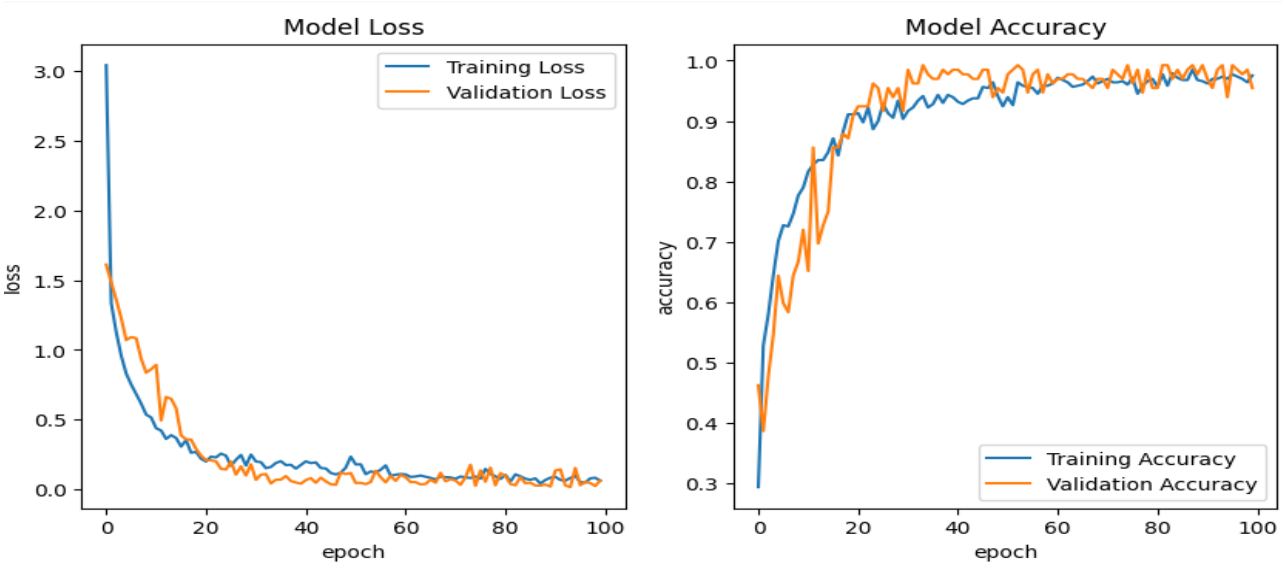


Figure III.25: Accuracy and losses graphs for training and validation.

The results of the best model that gives high accuracy are mentioned in the table below (Table III-6)

Table III.4: the best result

	Accuracy	loss
Training	97%	0.07
Validation	99 %	0.002

b) Confusion Matrix

Figure III-26 represents the confusion matrix, it summarizes the results of predictions and allows to evaluate the performance of our classification model.

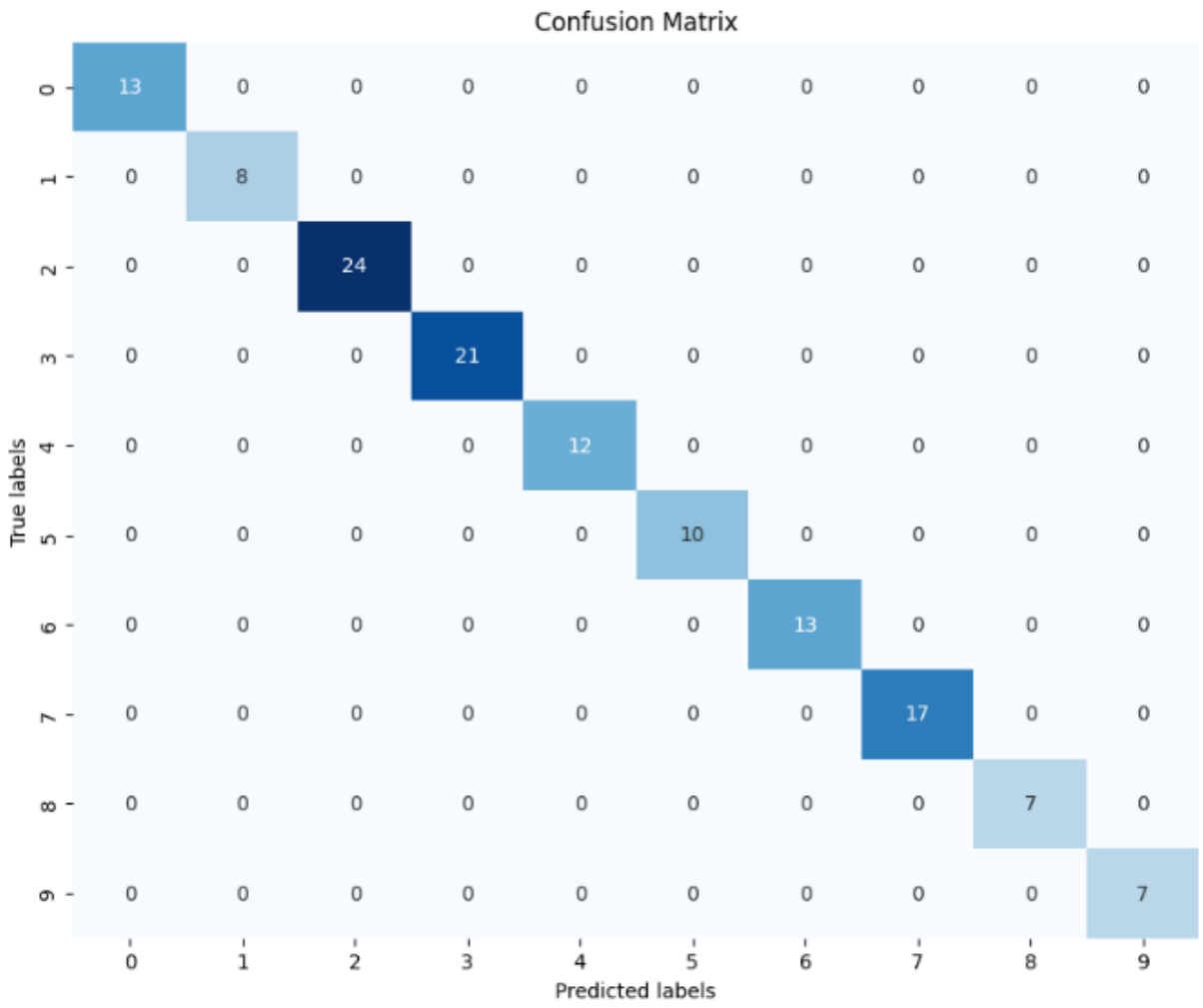


Figure III.26: confusion matrix.

III.8 Implementation

III.8.1 Model preparation

To implement the trained CNN model we need, as a first step, to converted the keras CNN model which is already saved as ".**h5**" format to ".**Tflite**" format to make it adaptable to be implemented in the ESP32, the figure III-27 represents the preparation of the model.

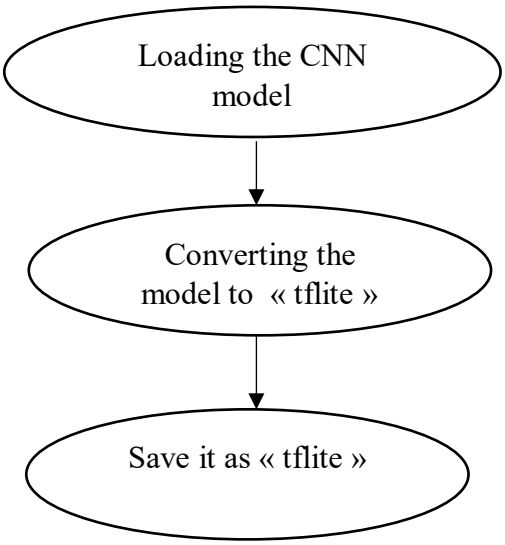


Figure III.27: the preparation model

Model conversion: we used the library **"tflite.TFLiteConverter.from_keras_model"** to convert the keras model from the form **"h5"** has the form **"tflite"** to be adapted to the esp32.

Saving the model: we opened a new file as form **"tflite"** and saved the converted model in this file.

III.8.2 Implementation result

After completing the data preparation phase and loading our trained model, we proceeded to test its performance in a real-world scenario. Figure III.28.

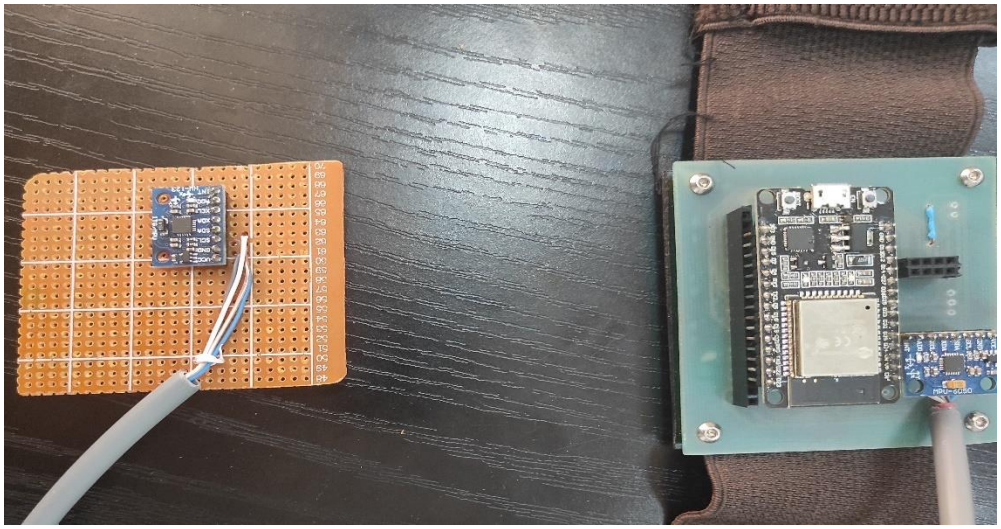


Figure III.28: the final system.

To implement the model on an ESP32 microcontroller, we utilized the Arduino IDE and a programmer. The following organigram outlines the steps involved in this implementation process:

- Begin the program execution.
- Set up the ESP32 microcontroller and the required peripherals.
- Configure the communication protocol, such as I2C, to enable data exchange between the ESP32 and the sensors.
- Initialize the sensors MPU6050.
- Initialize the model and load its parameters and weights (Model.h).
- Enter the main loop of the program.
- Read the sensor data from the MPU6050.
- Pre-process the sensor data to ensure compatibility with the input requirements of the model.
- Feed the pre-processed data into the loaded model.
- Perform inference using the model to predict the activities being performed.
- Print the predicted activity on the serial monitor for real-time monitoring and evaluation.
- Send the predicted class via WIFI connection to the admin computer.

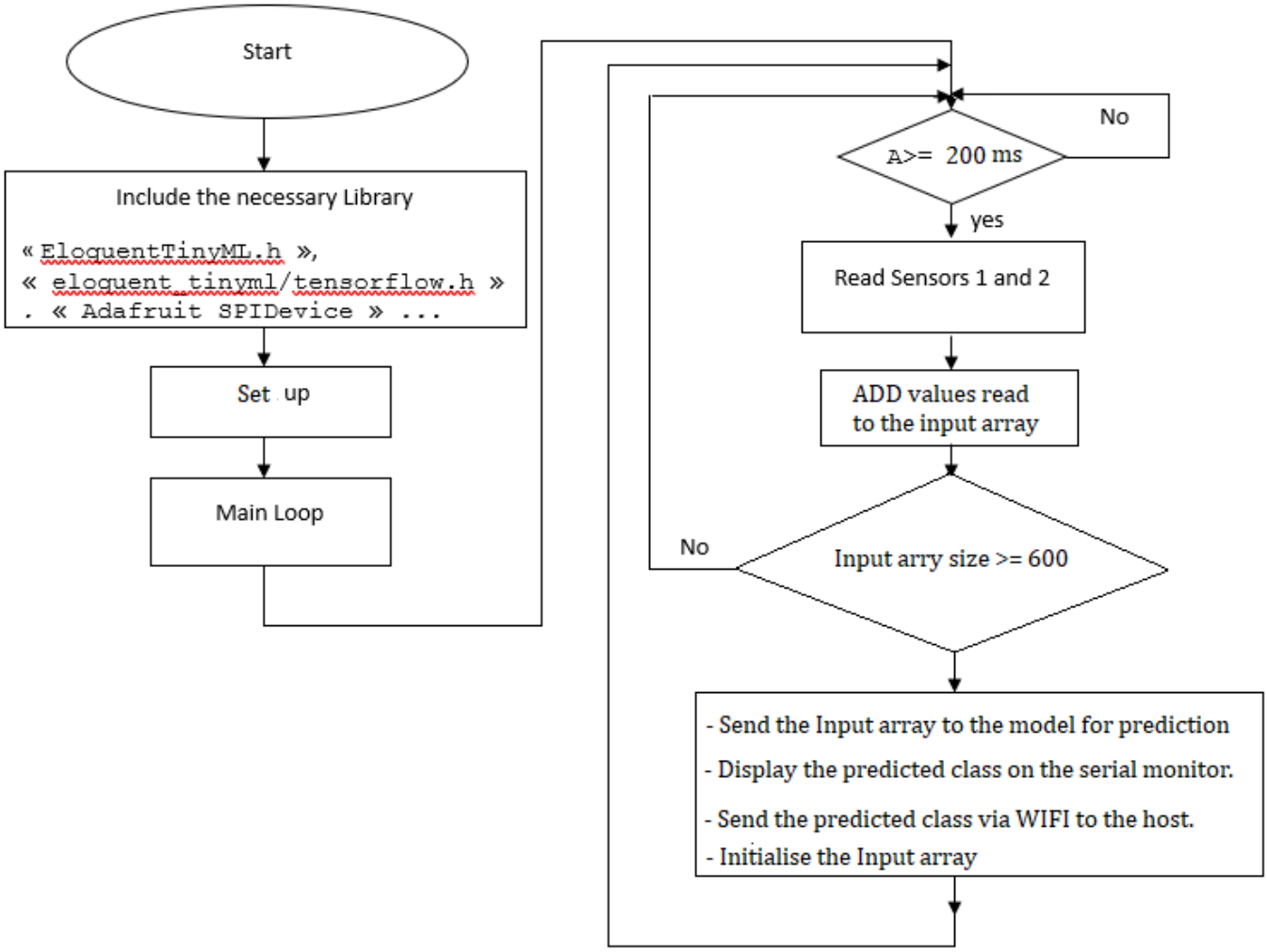


Figure III.29: Flowchart of ESP32 code.

By following this flowchart, we were able to successfully implement the model on the ESP32 microcontroller and observe the results in the serial monitor (figure III.25). This allowed us to assess the model's accuracy and effectiveness in classifying activities based on real-time sensor data.

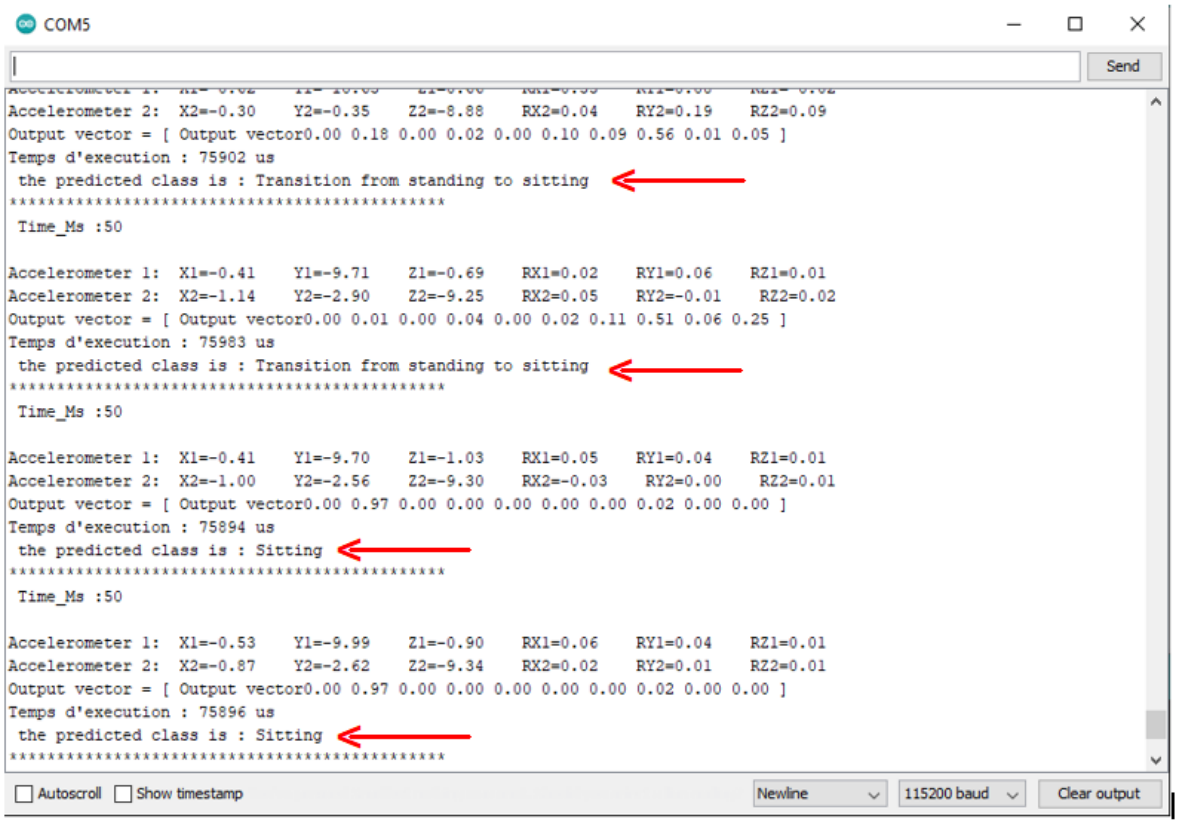


figure III.30: The results in the serial monitor

III.9 conclusion

This Chapter introduces the key concepts and components used in the project. The project aims to develop a system for monitoring the daily living activities of elderly individuals using IoT devices, sensors and AI technology. The components used include the ESP32 microcontroller and MPU6050 sensors. The ESP32 is a versatile microcontroller with built-in Wi-Fi and Bluetooth capabilities, while the MPU6050 is a sensor module that combines a gyroscope and accelerometer. The ESP32 communicates with the MPU6050 sensors using the I2C protocol.

Chapter III provides an overview of the project's architecture, including the connections between the components and the communication protocol. This chapter sets the foundation for the subsequent chapters, which will delve into the data collection process, data processing and analysis, and real-time monitoring of activities. Overall, Chapter III provides a comprehensive understanding of the project's components, their functionalities, and their interactions, setting the stage for the implementation and development of the system.

General conclusion

General conclusion

This project focuses on leveraging IoT technologies to monitor and assist the daily living activities of the elderly population. With the integration of ESP32 microcontroller, MPU6050 sensors, and wireless communication protocols such as Wi-Fi, real-time monitoring and assistance can be achieved.

The project builds upon the advancements in Artificial Intelligence (AI), particularly in the area of Convolutional Neural Networks (CNNs), to analyse the acceleration and rotational movements experienced by elderly individuals during their daily activities. By utilizing CNN models, the collected data is processed and classified into different activities, enabling effective monitoring and support.

The first chapter provides a comprehensive understanding of AI, its history, and the significance of ANN and CNN architectures. This knowledge serves as the foundation for implementing AI algorithms in the project.

The second chapter covers the basics of wireless sensor networks and IoT, including their definitions, applications, and architecture. It also addresses the various communication protocols used in WSN and IoT systems, as well as the importance of energy efficiency and the challenges related to security and privacy.

In the third chapter, the project's system components, including both hardware and software, are comprehensively introduced. The data collection process is thoroughly explained, and the collected data is meticulously prepared for training, validation, and testing purposes using the CNN model (1D). and implementing this approch in a small embedded system such as ESP32 was a real challenge.

The results obtained from these endeavours are exceptionally satisfactory, representing a significant milestone that paves the way for further advancements in this field. This project presents a highly promising approach to monitor and assist individuals living alone at home by leveraging the remarkable capabilities of IoT and AI technologies. Through the provision of real-time insights and support, it possesses the potential to significantly enhance the well-being

and safety of the elderly population, empowering them to maintain their cherished independence and quality of life.

References:

[1] S.C.Huang and T.-hieu Principles and Labs For Deep Learning , 2021.

[2] Chris Smith. The History of Artificial Intelligence. History of Computing, page 5, December 2006. https://courses.cs.washington.edu/courses/csep590/06au/projects/history-ai.pdf

[3] java point, history-of-artificial-intelligence, https://www.javatpoint.com/history-of-artificial-intelligence

[4] W. Ertel. Introduction to Artificial Intelligence, page 6,2nd Edition International Publishing AG 2017.

[5] Reynoso, R. (2021, May 25). A Complete History of Artificial Intelligence. Retrieved from G2: https://www.g2.com/articles/history-of-artificial-intelligence .Accessed Date: February 2023

[6] "What is the history of artificial intelligence (AI)?" Tableau URL: https://www.tableau.com/data-insights/ai/history Accessed: February 2023.

[7] http://en.wikipedia.org.

[8] AI Approches (2022, août,1) https://www.coursera.org/lecture/digital-transformation-course/ai-approaches-9vXdx , 1 août 2022

[9] VTU Pulse. Approaches to artificial intelligence. https://www.vtupulse.com/artificial-intelligence/approaches-to-artificial-intelligence/

[10] Massey University https://www.massey.ac.nz/~mjjohnso/notes/59302/l01.html . Accessed Date: February 2023

[11] Introduction to AI. A Modern approach, university of California, Berkeley, http://people.eecs.berkeley.edu/~russell/intro.htm . Accessed Date: February 2023.

[12] Tech Target. what is artificial intelligence

https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence

[13] https://www.ibm.com/topics/machine-learning

References:

[14]https://www.geeksforgeeks.org/supervised-unsupervised-learning/ 20 Avril 2023, Supervised and unsupervised learning)

[15] François Chollet DEEP LEARNING with Python 2018 by Manning Publications, page 94

[16] Ian Goodfellow Yoshua Bengio Aaron Courville Deep Learning page 25

[17] https://www.researchgate.net/figure/The-main-types-of-machine-learning-Main-approachesinclude-classification-and_fig1_354960266

[17] https://mindmajix.com/artificial-neural-network-and-how-it-works

[18]https://www.researchgate.net/figure/Representation-graphique-dun-reseau-de-neurones-Yilmaz-et-al-2015_fig3_343268354

[19]]https://mindmajix.com/artificial-neural-network-and-how-it-works

[20]https://www.superannotate.com/blog/activation-functions-in-neural-networks 2023

[21]: https://mlnotebook.github.io/post/CNN1/

[21] Université de 8 Mai 1945 – Guelma, Octobre 2020, Dr. Ferrage Med Amine

Présenté par : Hamouda Djallal Un système de détection d'intrusion pour la

Cybersécurité

[22] W. Ertel. Introduction to Artificial Intelligence, page 6,2nd Edition

International Publishing AG 2017.

[23]: MLNotebook. (n.d.). Convolutional Neural Networks (CNN) - An Illustrated Explanation. Retrieved from https://mlnotebook.github.io/post/CNN1/

[24] Fischione, C. (2014, September). An Introduction to Wireless Sensor Networks (Draft, Version 1.8, p. 15)

[25] Ameen, M. A., & Kwak, K. (2011, January). Social Issues in Wireless Sensor Networks with Healthcare Perspective. The International Arab Journal of Information Technology, 8, Inha University, Korea. Retrieved from https://www.researchgate.net

[26] Shutterstock. (n.d.). Images libres de droits de Capteur température. Retrieved from https://www.shutterstock.com/fr/search/capteur-temp%C3%A9rature

References:

[27] Grisoni, A., & Madelenat, J. (2021, December 22). Véhicule autonome, l'utopie à marche forcée. Mouvements. Retrieved from https://mouvements.info/vehicule-autonome-lutopie-a-marche-forcee/

[28] IFP Énergies Nouvelles. (2019, January 7). Smart City : les enjeux énergétiques de la ville durable. Retrieved from https://www.ifpenergiesnouvelles.fr/article/smart-city-les-enjeux-energetiques-ville-durable .

[29] ZigBee Specification. ZigBeeAlliance2006.http://www.zigbee.org/

[30] Vel Tech Rangarajan and Dr. Sagunthala. (2015, May 6-8). International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Chennai, India, pp. 418-423.

[31] Fei Lu, Guo-Hui Tian. (2012, April 25). The ZigBee Based Wireless Sensor and Actor Network in Intelligent Space Oriented to Home Service Robot. Retrieved from https://www.scirp.org/html/6-9701558/b8d22cb5-63ed-4701-a95c-7aaaffe28eab.jpg

[32] Mary K. Pratt. (2022, March 30). Top 12 most commonly used IoT protocols and standards. Retrieved from https://www.techtarget.com/iotagenda/tip/Top-12-most-commonly-used-IoT-protocols-and-standards

[33]“ "IoT Technology & Protocols – 7 Important IoT Communication Protocols." (May 2016). Retrieved from https://data-flair.training/blogs/iot-technology/

[34] "Utilisation du Bluetooth." (December 24, 2020). Retrieved from https://docwiki.embarcadero.com/RADStudio/Sydney/fr/Utilisation_du_Bluetooth

[35] Types of Wireless Communication Protocols in IoT." (September 2, 2019). Retrieved from https://iotdesignpro.com/articles/different-types-of-wireless-communication-protocols-for-iot

[36] https://www.researchgate.net/figure/An-Internet-of-Things-IoT-application-Smart-Life-scenario_fig1_287406143 , Design and Implementation of an IoT Access Point for Smart Home, Dec 2015, Tzu-Chia Wang, Jian-Cheng Chen's, Chin-Hwa Kuo's , Chih-Yung Chang

[37] Banti, K., Karampelia, I., Dimakis, T., Boulogeorgo, A.-A. A., Kyriakidis, T., & Louta, M. (2022). LoRaWAN Communication Protocols: A Comprehensive Survey under an Energy Efficiency Perspective. *Telecom*, 3(2), 18. https://doi.org/10.3390/telecom3020018

References:

[38] Gebhardt, P. (2019, December 18). What's LoRa again? And why is that relevant in terms of monitoring? *Paessler Blog*. Retrieved from https://blog.paessler.com/whats-lora-again.

[39] "PieEmbSysTech." Message Queuing Telemetry Transport (MQTT) Protocol. Retrieved from https://piembsystech.com/message-queuing-telemetry-transport-mqtt-protocol/.

[40] Khriji,S. Rym, Ch. (2022, December) Olfa KanounDynamic "Voltage and Frequency Scaling and Duty-Cycling for Ultra Low-Power Wireless Sensor Nodes". Tunisia.

[41] Zeyu, Zh. Lei Sh, Chunsheng Zhu3, & Mithun, M, "A Short Review on Sleep Scheduling Mechanism in Wireless Sensor Networks" Canada

[42] Rana, B., & Singh, Y. (2021, May). Duty-Cycling Techniques in IoT: Energy-Efficiency Perspective. Retrieved from https://www.researchgate.net/publication/353307566_Duty_-Cycling_Techniques_in_IoT_Energy-Efficiency_Perspective

[43] Zhou, M., Chen, H., Shu, L., & Liu, Y. (2022, July 1). UAV-Assisted Sleep Scheduling Algorithm for Energy-Efficient Data Collection in Agricultural Internet of Things. IEEE Internet of Things Journal, Retrieved from https://ieeexplore.ieee.org/abstract/document/9606182

[44] "ARTICLE Energy Harvesting for IOT – Status and Evaluation", Date: 29-06-2022, Place: Edmonton, Canada.

[45] Mustapha, I., Ali, B. M., Rasid, M. F. A., Sali, A., & Mohamad, H. (2015). An Energy-Efficient Spectrum-Aware Reinforcement Learning-Based Clustering Algorithm for Cognitive Radio Sensor Networks. Sensors,. doi:10.3390/s150819783, https://doi.org/10.3390/s150819783.

[46] B. Omodunbi, O.T. Arulogun, J.O. Emuoyibofarhe, "A review of energy conservation in wireless sensor networks" , pp. 17-23, 2013. Google Scholar

[47] "Ultra-low power techniques in energy harvesting wireless sensor networks", Scientific African Volume 11, March 2021.

[48] D. Sathya, P. Kumar"Secured data aggregation in wireless sensor networks",pp. 369-375, 10.1108/sr-06-2017-0103, 2018.[Google Scholar]

[49] "Ultra-low power techniques in energy harvesting wireless sensor networks: Recent advances and issues",https://doi.org/10.1016/j.sciaf.2021.e00720, Volume 11, March 2021.

References:

[50] H. Harb, C.A. Jaounde, A. Makhoul, (2020), "An energy-efficient data prediction and processing approach for the internet of things and sensing based applications", Peer-peer Netw. pp. 780-795.

[51] Dheyab Salman..Fadhil Zaidan, "Routing Protocols for Energy Efficiency in WSNs": A review, https://www.researchgate.net/publication/350300373 . March 2021.

[52] S. Jaydip (2009). A Survey on Wireless Sensor Network Security Journal: International Journal of Communication Networks and Information Security Volume: 1 Issue: 2 pages:55-78

[53] M. A. Burhanuddin1, Ali Abdul-Jabbar Mohammed1, Ronizam Ismail2, Mustafa Emad Hameed3, Ali Noori, Kareem4 and Halizah Basiron1 "A Review on Security Challenges and Features in Wireless Sensor Networks: IoT Perspective"

[54]"3 Principles of Infosec: The CIA Triad";https://carbidesecure.com/resources/3-principles-of-infosec-the-cia-triad/

[55] https://www.researchgate.net/figure/ESP32-functional-block-diagram_fig5_341446512

[56]http://emery.claude.free.fr/arduino-lcd1602-i2c.html