PROJET 2

A Fuzzy Logic based Offloading System for Distributed Deep Learning in Wireless Sensor Networks



Prepared By

Abdoul Bassit NKOUONJOM

Nunike Ayetei LAMPTEY

VINCENT COLIN

Supervised By

Carol HABIB

Academic Year 2022-23

SUMMARY

I. Introduction

A fuzzy logic based offloading system for distributed deep learning in wireless sensor networks is a system that uses fuzzy logic to make decisions about which parts of a deep learning model should be executed on the edge devices (sensors)(1) and which should be offloaded to a remote server.

Fuzzy logic is a mathematical approach that uses linguistic variables and fuzzy rules to represent and manipulate imprecise or uncertain data. In a distributed deep learning system, fuzzy logic can be used to determine the optimal offloading strategy based on factors such as the computational capabilities of the edge devices, the amount of data to be processed, the network bandwidth and latency, and the energy consumption.

The system can be implemented in a distributed manner, with each edge device running a fuzzy logic controller that makes local offloading decisions based on the current system state and its own resources. The controller can then communicate with other edge devices and the server to coordinate offloading and share partial results. This offloading is done through a process called collaborative inference(2).

We will address the case where edge devices are sensor nodes that form a wireless sensor network (WSNs)(3). WSNs are composed by nodes ranging from few tens to over thousands. Each node communicates with other nodes depending on his radio coverage area.[1]

WSNs(see Annex below) extract several types of data including multivariate dynamical time series data. These data are processed by DNNs and especially Recurrent Neural Networks – Long-short Term Memory(RNN-LSTM)(4), which are the most suitable neural networks architecture for time series data analysis[1]

By using fuzzy logic, the system can adapt to changing conditions and make efficient use of the available resources, improving the performance and reliability of the distributed deep learning system

II. Context

Internet of things (IoT)(5) devices have received a significant reception since its inception and have gradually become ubiquitous. This peak is a result of COVID-

19 pandemic and the global supply chain disruption. At the end 2022, Internet of things is expected to grow 18% to 14.4 billion active connections [1]. These devices are wildly used in diverse industries which include Agriculture, Healthcare, Transportation, and the energy industry. These devices rely on deep neural networks (DNNs)(6) to execute inference tasks.

Wireless Sensor networks which form part of the IoT community are used to capture data from various areas and are used widely to improve smart cities and Industries. This raw information captured is streamed from sensor node points into the centralized cloud systems(7). Data in these centralized cloud systems are used to make informed decisions based on predictions using advanced machine learning and deep learning methods. [1]

The increasing deployment of wireless sensor devices has led to the overwhelming amount of data collected over networks, this progress has caused an increase in the cost of communications between sensors and centralized cloud systems. The repercussion of this is the waste of energy and saturation of network. There is the need to find a way to be able to find the energy level of sensor devices and the available band width at every instance to prevent waste of energy and increase the efficiency of wireless sensor devices.

III. Problematics

WSNs are used in a variety of applications as said earlier, and as a result, some common problems where observed which include:

- Limited power and energy: Because WSNs are often battery-powered, energy efficiency is a key concern. Sensor nodes must use energy efficiently in order to extend the lifetime of the network.
- Limited communication range: The communication range of wireless sensor nodes is typically limited, which can make it challenging to transmit data over long distances or through obstacles.
- Limited computing resources: WSNs often use small, low-power devices that have limited computing resources. This can make it challenging to perform complex tasks or process large amounts of data.
- Security: WSNs are vulnerable to security threats such as hacking and spoofing. Ensuring the security of data transmitted over the network is an important concern.

- Interference: WSNs can be disrupted by interference from other wireless devices or sources of radio frequency (RF) noise.
- Data processing and analysis: WSNs generate large amounts of data, which can be challenging to process and analyze in real-time.

The aim of this project is to show that the implementation of our controller on edge devices (smart sensor nodes) that run RNN-LSTM for multivariate time series predictions can reduce their energetic cost by around 50%.[1]

IV. Strategy and Feasibility Studies.

A. Objectives

Our main objective in this project is to prove and implement on a practical level that our fuzzy logic controller on edge devices (smart nodes) that run RNN-LSTM for multivariate time series predictions :

- can be used to optimize the allocation of tasks in a distributed deep learning(8)system, leading to improved performance and faster completion times.
- can be used to process sensor data and make more accurate predictions, leading to better results in deep learning applications.
- can be used to optimize the use of resources in a distributed deep learning system, leading to reduced energy consumption and longer battery life for sensor nodes.

Thus, reducing their energetic cost by around 50%.

B. Technical Feasibility

First thing first we will need to setup the network architecture of our model which are the Edge level, the fog level, and the cloud(Architecture in appendix)

1. Edge Level

This is the level where all the edge computing and collaborative inference is handled. In our work, the Raspberry Pi(9) will be used as our Sensor Nodes{SN_i}(10).

The Raspberry pi will be setup using a MicroSD card with a Linux distribution(Ubuntu)(8) to manage the system, a 5V battery as power source and a python(11) script with the fuzzy logic control that will guide the Sensors decisions to perform offloads according to the scenario chosen. At this level, we will also collect three{3} important variables which are a multivariate Time series data type. These variables will serve as input for our logic controller. These Variables are:

a. Data Availability rate

The Sensor node's main function is to collect data from its surrounding. This could be things like temperature readings, humidity levels or pressure measurements. To get this variable, we will write a python function that monitors the actual data collected over time and compare it to the expected data collection or expected data threshold. This will allow us to see how closely the actual data collection matches the expected data collection and will give us an idea of the data availability of the sensor node.

b. The Network Bandwidth percentage

This refers to the amount of data that can be a transmitted in a network at a given time. When it comes to this project, the most appropriate technology to connect the Sensor nodes(raspberry Pi) to each other to simulate real lifetime scenario will be Wi-Fi[1]. We see that the nodes are all connected to a gateway(From the WSN figure in appendix).

This gateway(12) will be a small mobile WI-FI router to which all the SNs will be connected. We can make the sensor nodes to have the information of the network state by writing a python code which will get the active transmitting and receiving nodes from the gateway and comparing the used bandwidth over the network bandwidth, thus obtaining the Network bandwidth percentage available as input in our fuzzy logic model. This is a theoretical approach that would be confirmed when realizing the project since we couldn't find a more appropriate approach during the research phase.

c. The energy Consumption

Our aim here will be to let know the SNs the amount of energy left from its power supply. The Raspberry Pi will be connected to a 5V battery as power supply. Since the Capacity of a battery varies, we intend to use a 4200mAh rechargeable(see figure below) battery. To get the energy consumption, we will connect a RDTech UM35C[2] module(see figure in appendix) between the Battery supply and the Raspberry pi which function will be to measure the actual power at a given time. We can then get a table of the energy consumption through a python script that we found on Github[3] for the RDtech UM35C module.

For the SN to be aware of the energy left, we will write a python script that will take the power left from the generated table and used it as input for our fuzzy controller.

2. The fog Level

A decision will have to be taken at the level of SNs to maximize their internal energy while preserving the network from saturation. For this, each SN will be equipped with the proposed offloading model in order for it to offload its inferences tasks either horizontally to another SN or vertically to the upper level[1] (as depicted in Figure in appendix below).

Fuzzy logic uses a mathematical approach that allows a system to make decisions based on uncertain or imprecise data. It is often used in control systems to determine the appropriate action to take in each situation.

To design a fuzzy logic controller, we will follow these steps:

- Identify the input variables and output variables of the system. The input variables are the factors that influence the system's behavior(the Data availability, the Network Bandwidth, and the Energy Consumption), while the output variables are the actions taken by the Sensor Nodes.
- Define the membership functions for each input variable above. A membership function is a curve that defines the range of values for an input variable and the degree to which each value belongs to that variable. (Membership function in annex)
- Define the rules for the system. These rules should specify how the input variables influence the output variables.

- Implement the fuzzy logic controller using a programming language such as Python.
- Test and tune the controller to ensure that it is performing as desired. This may involve adjusting the membership functions or rules to achieve the desired behavior.

To see if our controller will be effective, we have experimental values. In these experimental values we have 3 different scenarios for the experiments[1].

Scenarios:

- Scenario 0: we don't use the controller and we are letting each sensors node make its inferences
- Scenario 1: Consists in giving the possibility to the sensors node to offload their inferences vertically at the upper level
- Scenario 2: Consists in giving the possibility to the offload their inference vertically to the upper level or horizontally to another sensor node

3. The Cloud

The cloud here refers to the platform where the processed collected data will be finally uploaded to for storage. In our project, a simple personal computer will be used as the storage facility.

C. Economic feasibility

In need to carry out our project in good conditions, we came out with a list of the necessary tools to be used as follows:

- A Raspberry Pi (Sensor Node)
- Dual 18650 Lithium Battery Shield V8[4]
- A set of batteries (4200mAh)
- A personal Computer(LAPTOP)
- Wi-Fi router (the same Wi-Fi between laptop and Raspberry Pi)
- RDTech UM34C USB 3.0 Type-C DC Voltmeter
- Web GUI for RDTech USB testers [3]
- Software and language code (python)

After research it came out that we had some of the tools available from our personal belongings and we will need :

- Dual 18650 Lithium Battery Shield V8
- A set of batteries (4200mAh)
- RDTech UM34C USB 3.0 Type-C DC Voltmeter
- Web GUI for RDTech USB testers (from Github)

A cost overview of the necessary tools will be as follows

Equipment	Quantity	Unit Price(€)
Dual 18650 Lithium battery shield	2	2,4
4200mAh Batteries	1	14,97
RDTech UM34C	2	15
Total	5	49,77

D. Observed Limitations

We did not find the project theme quite difficult as the theme was clear, and the theoretical part of the project had already been carried out. However, there may also be challenges to consider when implementing a fuzzy logic based offloading system for distributed deep learning in wireless sensor networks, including:

- Complexity: Fuzzy logic algorithms can be complex to implement and may require specialized knowledge and expertise.
- Limited resources: Wireless sensor networks may have limited resources such as computational power and bandwidth, which could impact the performance of a fuzzy logic based offloading system.

- Interference: Wireless sensor networks may be subject to interference from other devices or environmental factors, which could impact the accuracy and reliability of the system.

Our challenge will be to carry out the project considering these setbacks and trying our best to produce results closely enough as those found in the theoretical simulation.

V. Project Management

This project is a quite straight forward project with all the variables being linked to each other. There are several project management methods that we can use to plan, execute, and monitor the progress of a project, but the best approach for the project is the Waterfall method. The waterfall method is a linear approach to project management in which each phase of the project is completed before moving on to the next phase. This method is typically used for projects with well-defined requirements and a clear end goal as it is the case in our project.

The key stages of the waterfall method are:

- Requirements definition: The first step in the waterfall method is to clearly define the project requirements and scope. This is what we did in the first part of our research. We clearly defined our goal and project requirements to achieve that.
- Design: In the design phase, we created a detailed plan for how the project will be implemented and summarized it in a Gantt chart(see annex).
- Implementation: In the implementation phase, we are to build and test the project according to the design plan. This will involve writing python code, building prototypes, or creating other deliverables.
- Testing: The testing phase involves verifying that the project meets the requirements and functions as intended. This involves conducting formal tests or conducting user acceptance testing to ensure that the project meets our desired goal.

- Maintenance: The final step in the waterfall method is ongoing maintenance and support to ensure that the project continues to meet the needs and ensure that bugs are controlled.

We equally designed a Trello board(See figure in Annex) to share the workload among all three(3) members to easily accomplish the necessary task.

VI. Conclusion

Throughout this report, we gave a brief overview of our project goal, which is to practically demonstrate that adding a fuzzy logic controller to a sensor node with input : Data availability, network bandwidth and Energy consumption, can lower the sensor nodes energy consumption by 50% through collaborative inference. The project is an innovative project, with a game changing approach in the world of internet of things(IOT). It is a great challenge for us to tackle such a great project. We feel confident in our ability and technical skills to go through the project, and we are excited to make it come to reality.

VII. Appendix

A. Definition of Terms

Edge devices *: An edge device is a type of computer that is located at the boundary of a network and serves as a communication and control point between the network and other devices. Edge devices are often used to connect devices to the internet or to perform tasks such as data collection, processing, and analysis. They are designed to operate in challenging environments, such as in remote or industrial locations, and are often ruggedized to withstand extreme temperatures, dust, and vibration.*

Wireless Sensor Networks *: Wireless sensor networks (WSNs) are networks of small, inexpensive, and low-power devices that are equipped with sensors, processors, and wireless communication capabilities. These devices are used to monitor and collect data from the physical world and transmit the data back to a central location for analysis. WSNs are used in a variety of applications, such as environmental monitoring, industrial control, and security.*

Edge computing *: It is a distributed information technology (IT) architecture in which client data is processed at the periphery of the network(the Sensor Nodes in our case), as close to the originating source as possible. ("What is edge computing? Everything you need to know - SearchDataCenter")*

Collaborative inference*: The aim of collaborative inference between the edge-fog-cloud infrastructure is to enable local and remote inference without degrading the accuracy of DNNs models. The key of an efficient collaboration relies on an a good internal and external communication between multi-level devices present in the hierarchy*

RNN-LSTM: *Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input.* ***("Understanding RNN and LSTM. What is Neural Network? - Medium")***

Internet of Things(IOT) *: The Internet of Things (IoT) describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. ("What Is the Internet of Things (IoT)? - Oracle")*

Deep Neural Networks*: A deep neural network (DNN) is a type of artificial neural network (ANN) that is composed of many layers of interconnected neurons. It is called "deep" because it has a large number of layers, typically consisting of hundreds or thousands of neurons. Deep neural networks are used for tasks such as image and speech recognition, natural language processing, and machine translation.*

Distributed Deep Learning *: Distributed deep learning is a type of deep learning that involves training a deep neural network across multiple machines, either in a single location or in multiple locations connected over a network. The goal of distributed deep learning is to train larger and* *more accurate models than would be possible on a single machine, by leveraging the additional computational resources and data available.*

Centralized Cloud systems *: Centralized cloud systems are computing environments that are based on the cloud computing model, in which resources such as computing power, storage, and software are provided as services over the internet. In a centralized cloud system, these resources are typically provided by a single provider, such as a large tech company, and are accessed by users via the internet.*

The Raspberry Pi *: The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.*

Fuzzy controller Laws *: After declaring our input and output variables and associating membership functions, the next step consists of linking all the variable by building rules which will govern our controller. Our rule database is composed by 27 rules. Below is an example of a rule to highlight the format of the rules: If (energy is good AND bandwidth is poor AND data is good) Then (offload is No).*

Python *: Python is a high-level, interpreted programming language that is widely used for web development, scientific computing, data analysis, and artificial intelligence. It was created in the late 1980s by Guido van Rossum and has a large, active community of users and developers.*

One of the key features of Python is its simplicity and readability. It uses a clear and concise syntax, which makes it easy to learn and understand. Python also has a large standard library, which includes many modules and packages for common programming tasks, such as connecting to web servers, reading, and writing files, and working with data.

Gateway *: a gateway is a device that connects the sensor nodes to a central location or to the internet. The gateway acts as a bridge between the WSN and other networks, and is responsible for collecting data from the sensor nodes, performing any necessary processing or aggregation, and forwarding the data to the appropriate destination.*

VIII. Figures and Tables.

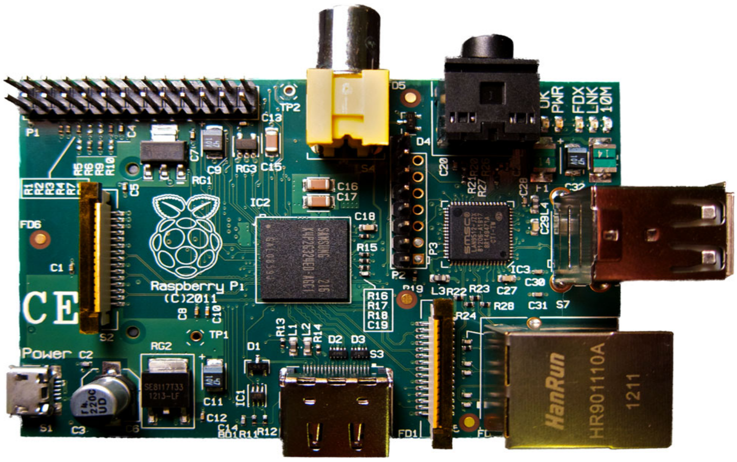
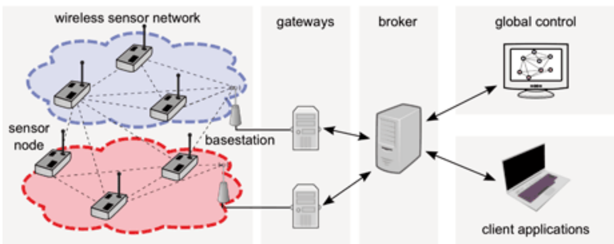


Figure 1 : A Raspberry PI



"Cette photo par Auteur inconnu est soumise à la licence CC BY" ("Cette photo par Auteur inconnu est soumise à la licence CC BY -NC -ND")

Figure 2 : A wireless Sensor Node

UM34C Log Data Graph Connected COM100 Name testing2 Rate 1s Disconnect

Session name current Select Export Destroy

Time	Voltage	Current	Power	Temperature	Data +/-	Mode	Accumulated	Resistance
2018-10-28 13:55:58	5.06 V	0.208 A	1.052 W	27.0 °C	+0.01 / -0.0 V	DCP1.5A	212 mAh / 1073 mWh / 3676 seconds	24.3 Ω
2018-10-28 13:56:00	5.05 V	0.208 A	1.05 W	27.0 °C	+0.01 / -0.0 V	DCP1.5A	212 mAh / 1074 mWh / 3678 seconds	24.2 Ω
2018-10-28 13:56:01	5.06 V	0.207 A	1.047 W	27.0 °C	+0.01 / -0.0 V	DCP1.5A	212 mAh / 1074 mWh / 3679 seconds	24.4 Ω
2018-10-28 13:56:02	5.06 V	0.208 A	1.052 W	27.0 °C	+0.0 / -0.0 V	DCP1.5A	212 mAh / 1075 mWh / 3680 seconds	24.3 Ω
2018-10-28 13:56:03	5.06 V	0.208 A	1.052 W	27.0 °C	+0.01 / -0.0 V	DCP1.5A	212 mAh / 1075 mWh / 3681 seconds	24.3 Ω
2018-10-28 13:56:05	5.05 V	0.208 A	1.05 W	27.0 °C	+0.0 / -0.0 V	DCP1.5A	212 mAh / 1075 mWh / 3683 seconds	24.2 Ω
2018-10-28 13:56:06	5.06 V	0.207 A	1.047 W	27.0 °C	+0.0 / -0.0 V	DCP1.5A	212 mAh / 1076 mWh / 3684 seconds	24.4 Ω
2018-10-28 13:56:07	5.06 V	0.208 A	1.052 W	27.0 °C	+0.0 / -0.0 V	DCP1.5A	212 mAh / 1076 mWh / 3685 seconds	24.3 Ω
2018-10-28 13:56:09	5.06 V	0.208 A	1.052 W	27.0 °C	+0.01 / -0.0 V	DCP1.5A	212 mAh / 1076 mWh / 3686 seconds	24.3 Ω
2018-10-28 13:56:10	5.06 V	0.207 A	1.047 W	27.0 °C	+0.01 / -0.0 V	DCP1.5A	212 mAh / 1077 mWh / 3688 seconds	24.4 Ω
2018-10-28 13:56:11	5.06 V	0.208 A	1.052 W	27.0 °C	+0.0 / -0.0 V	DCP1.5A	212 mAh / 1077 mWh / 3689 seconds	24.3 Ω

Figure 3 : Power consumption output table

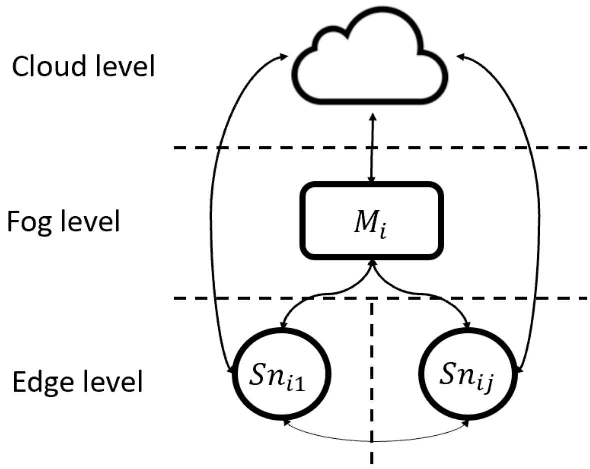


Figure 4 : Network Architecture of our model

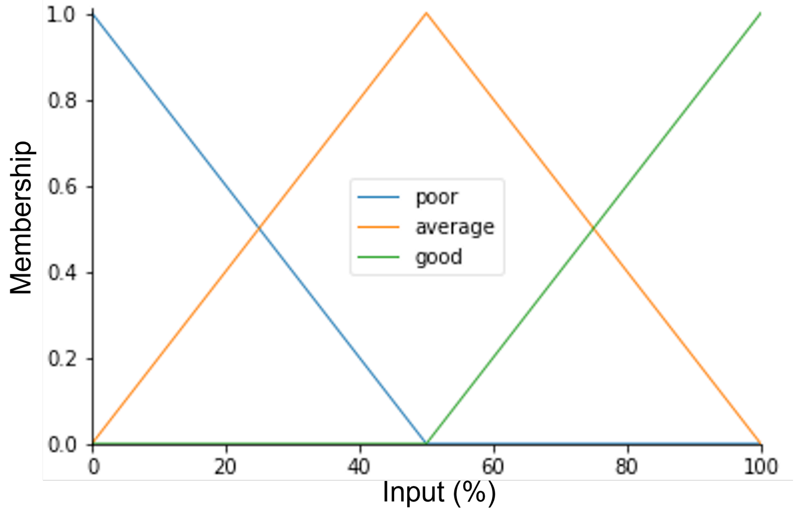


Figure 5 : Membership Function for Fuzzy logic

Figure 6: RDTech UM34C module for power measurements

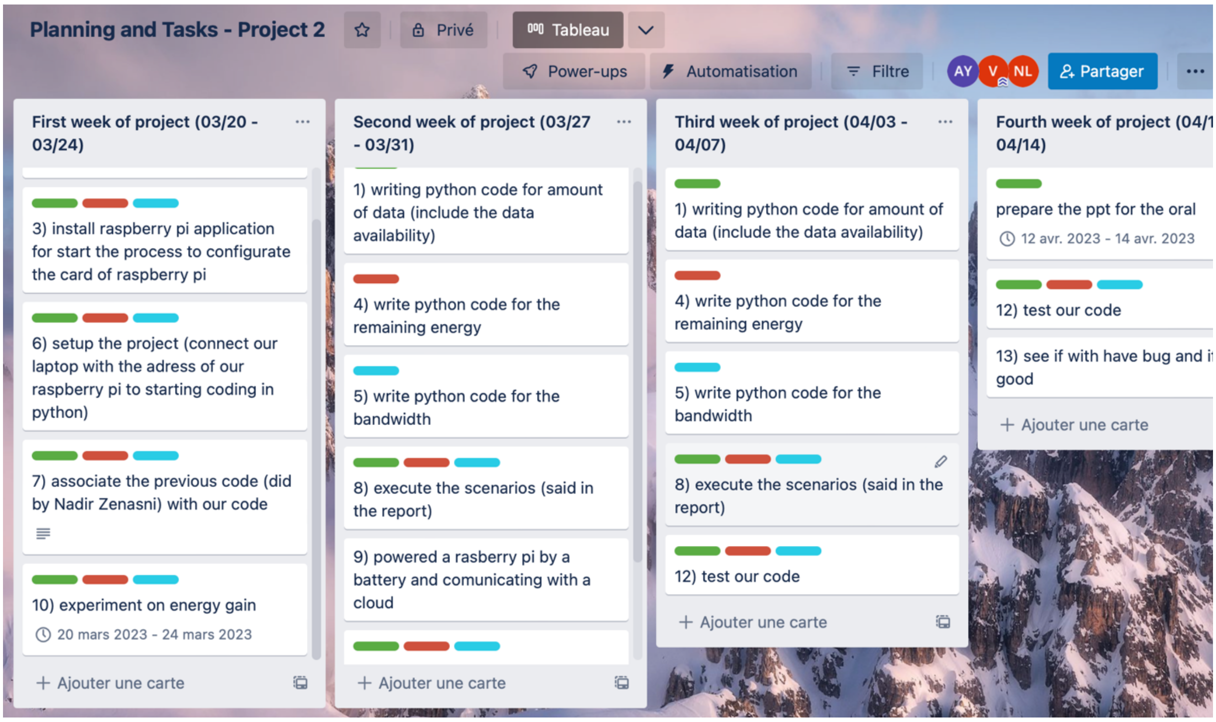


Figure 7 : Trello board

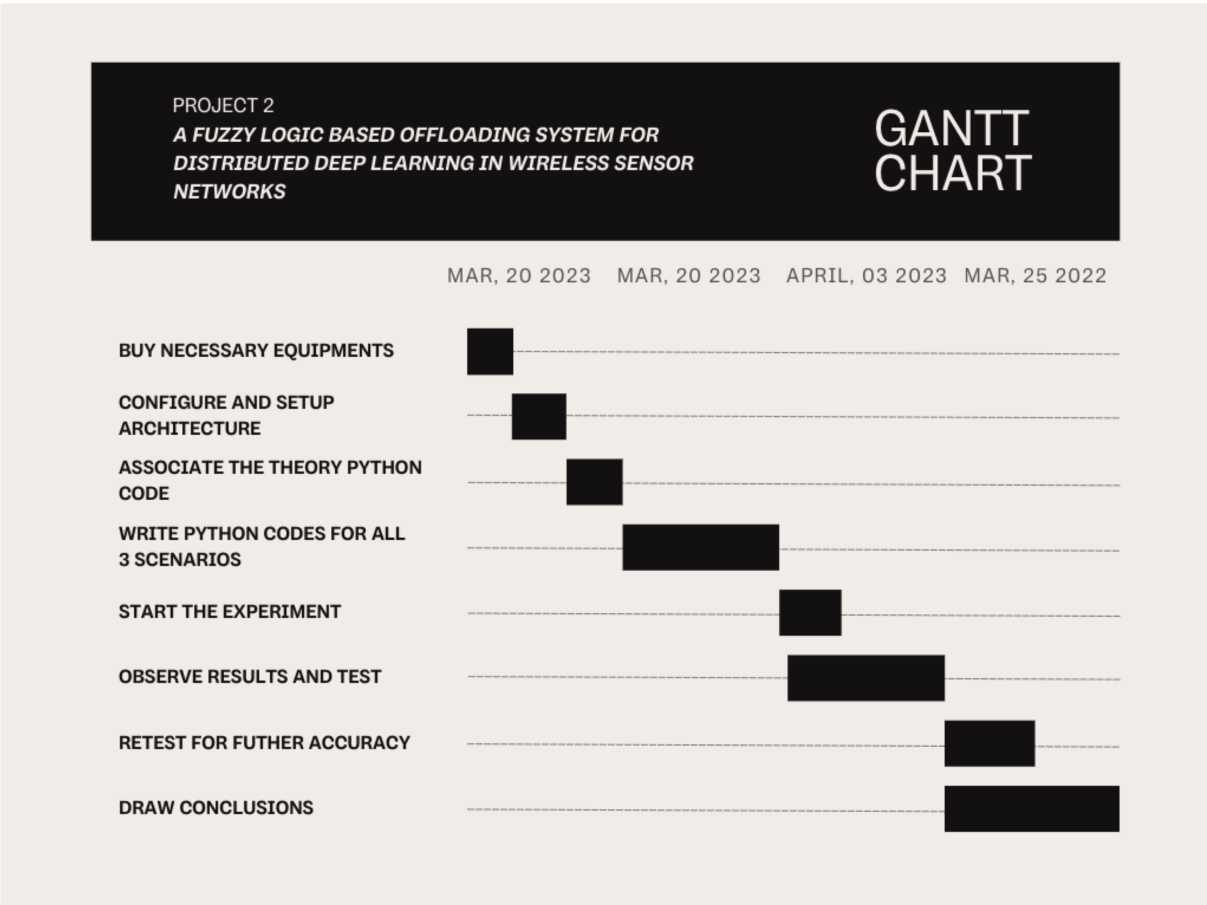


Figure 9 : Gantt chart

IX. References

[1] A Fuzzy Logic based Offloading System for Distributed Deep Learning in Wireless Sensor Networks Nadir Zenasni ∗, Carol Habib† and Jad Nassar‡

[2] https://github.com/smandon/rdumtool

[3] https://github.com/kolinger/rd-usb

[4] https://projetsdiy.fr/shield-batterie-18650-projets-iot-esp32-esp8266-raspberry-pi/

https://codecrucks.com/what-is-fuzzy-membership-function-complete-guide/

https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjk5NbS7P77AhW_aqQEHSM3A_wQFnoECAgQAw&url=https%3A%2F%2Flinuxhint.com%2F10-best-programming-languages-to-learn-on-raspberry-pi%2F&usg=AOvVaw1hPHCzuBs_Qpz1g8IThgB7

https://www.techtarget.com/searchdatacenter/definition/edge-computing

https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

https://www.oracle.com/internet-of-things/what-is-iot/

https://www.circuitbasics.com/how-to-power-your-raspberry-pi-with-a-lithium-battery/