



## Licence en Sciences et Techniques Département de Physique

**Filière : Electronique, Electrotechnique et Automatique**

**Prévision des incendies de forêt en se basant sur l'internet des  
objets et l'Intelligence artificielle**

### *Projet de fin d'études*

**Projet réalisé par :**

- ✓ Nouhaila Boudaibi
- ✓ Dounia Toullichi

**Dirigé par :**

- ✓ Pr. FAHIM Mohamed

**Soutenue le : 22 juin 2023, devant les membres de jury :**

- ✓ Pr. Mohammad ELGHABZOURI ..... Président.
- ✓ Pr. Oussama EL ABOUTI ..... Examinateur.
- ✓ Pr. FAHIM Mohamed ..... Encadrant.

## **Remerciements**

Avant d'entamer, nous tenons à remercier DIEU qui nous a facilité le chemin pour réaliser ce projet et nous a donné le courage et la patience pour surmonter les obstacles, les chutes et la déception.

Nous adressons également nos remerciements à notre encadrant Monsieur MOHAMED FAHIM pour son engagement indéfectible, son soutien inestimable et ses conseils éclairés tout au long de cette période.

Nous voudrons témoigner toutes nos reconnaissances à tous les professeurs qui ont contribué dans notre parcours scolaire, spécialement nos chers professeurs de la FSTH.

Nous tenons aussi à remercier les membres du jury qui ont accepté de juger notre modeste travail et de l'enrichir avec leurs remarques précieuses.

Un grand merci à nos chers parents pour leur soutien financier et moral, durant toutes ces années de sacrifice et de patience. Nos frères et sœurs pour les encouragements, leur support était un véritable moteur de motivation et de réussite.

Enfin, ils nous semblent indispensable de mentionner toute personne ayant apporté sa contribution de près ou de loin à la réalisation de notre projet.

## Dédicace

### **À ma merveilleuse mère,**

Tu es la source de ma vie, mon guide, ma confidente. Dans les hauts comme dans les bas, tu étais là pour moi, m'encourageant et me soutenant inconditionnellement.

### **À mon cher père,**

Bien que tu ne sois plus physiquement présent, ton esprit continue de m'inspirer chaque jour. Tu étais un modèle de force, de sagesse et de bienveillance. Je suis honorée d'être ta fille, de porter avec moi les principes que tu m'as inculqués.

### **À mes chers frères,**

Vous avez été mes piliers, mes sources constantes de motivation. Votre présence et vos paroles inspirantes ont nourri mon esprit et ont allumé en moi une flamme d'audace et de détermination.

### **À ma chère nièce,**

Tu es un rayon de soleil dans nos vies, apportant la joie et le bonheur à notre famille. Ton innocence et ta pureté d'esprit sont des atouts précieux me permettront de voir la beauté et la magie dans les petites choses de la vie.

### **À tous mes enseignants,**

Je suis reconnaissante des efforts inlassables que vous avez déployés pour m'offrir une éducation de qualité. Votre passion pour enseigner, votre patience et votre détermination à m'aider à réussir ont été des sources d'inspiration constantes.

***Nouhaila***

## Dédicace

### **À mes chers parents,**

Qui m'avez toujours encouragée. Ce modeste travail que je présente est le fruit de tous les sacrifices que vous avez consentis pour mon éducation et ma formation. Que Dieu vous protège et vous comble de ses bénédictions.

### **À mes chers frères,**

Qui avez toujours été mon soutien indéfectible dans tous les moments difficiles. Votre présence et votre soutien m'ont été d'une importance indispensable et je tiens à exprimer ma gratitude.

### **À ma chère sœur,**

Pour ta précieuse présence et ta compréhension. Ce que tu as apporté dans ma vie n'a pas de prix, je suis consciente que je ne pourrais jamais le rembourser. Merci d'être la sœur extraordinaire que tu es.

### **À mes amies et mes proches,**

Pour vos conseils, votre soutien et vos encouragements. Vous avez été présents tout le temps pour moi, me prodiguant un amour inconditionnel.

*Dounia*

## Résumé

Les incendies de forêt constituent un risque naturel majeur pour les forêts du monde entier, affectant à la fois les populations et les infrastructures. Dans ce rapport, nous proposons de concevoir un système intelligent de prévision des incendies de forêt en utilisant des technologies telles que le Machine Learning et l'IoT. Notre objectif principal est de prévoir ce phénomène en analysant les données météorologiques collectées par les capteurs, notamment le DHT11 et l'anémomètre. En utilisant des algorithmes de classification supervisés, nous serons en mesure d'anticiper les conditions propices à la survenue d'un incendie. De plus, la solution proposée est dotée d'un système d'alerte en cas de prévision d'un incendie, afin de permettre une réponse rapide et efficace.

**Mot clés :** Incendie de forêt, IOT, AI, Machine Learning, Classification, Prévision, Alerta.

## Abstract

Forest fires pose a major natural risk to forests around the world, affecting both populations and infrastructure. In this report, we propose to design an intelligent forest fire prediction system using technologies such as Machine Learning and IoT. Our main objective is to predict this phenomenon by analysing the meteorological data collected by the sensors, including the DHT11 and anemometer. By using supervised classification algorithms, we will be able to anticipate conditions conducive to the occurrence of a fire. In addition, the proposed solution is equipped with a fire warning system to allow for a quick and effective response.

**Keywords :** Forest fire, IOT, AI, Machine Learning, Classification, Prediction, Alert.

---

# TABLE DES MATIÈRES

Remerciements . . . . .	i
Dédicace . . . . .	ii
Dédicace . . . . .	iii
Résumé . . . . .	iv
Abstract . . . . .	iv
<b>Liste des abréviations</b>	<b>xii</b>
<b>introduction générale</b>	<b>1</b>
<b>1 Présentation et cadrage du projet (Cahier de charge)</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Présentation du sujet . . . . .	2
1.3 Contexte du projet . . . . .	3
1.4 Problématique . . . . .	3
1.5 Objectifs . . . . .	5
1.6 Etude de l'existant . . . . .	5
1.7 Produit du projet (Solution Proposée) . . . . .	9
1.8 Besoins fonctionnels . . . . .	9
1.9 Besoins non fonctionnels . . . . .	10
1.10 Déroulement du projet . . . . .	11
1.10.1 Planification opérationnelle . . . . .	11
1.10.1.1 Tableau des tâches . . . . .	11
1.10.1.2 Clarification des tâches . . . . .	11
1.10.1.3 Diagramme de GANTT . . . . .	12

1.10.2	Processus de développement . . . . .	13
1.10.2.1	Présentation du cycle de vie de développement . . . . .	13
1.10.2.2	Phases . . . . .	14
1.11	Conclusion . . . . .	15
<b>2</b>	<b>Généralités sur l'internet des objets et l'Intelligence artificielle</b>	<b>16</b>
2.1	Introduction . . . . .	16
2.2	C'est quoi l'internet des objets ? . . . . .	16
2.3	Intelligence artificielle . . . . .	18
2.3.1	Types de l'Intelligence artificielle . . . . .	18
2.3.2	Applications de l'Intelligence artificielle . . . . .	19
2.3.3	Différence entre Intelligence artificielle et Machine Learning et Deep Learning . . . . .	20
2.3.4	Machine Learning . . . . .	20
2.3.4.1	Machine Learning supervisé . . . . .	21
2.3.4.2	Machine Learning non supervisé . . . . .	23
2.3.4.3	Machine Learning par renforcement . . . . .	24
2.3.5	Deep Learning . . . . .	24
2.4	Conclusion . . . . .	25
<b>3</b>	<b>Conception et réalisation du prototype</b>	<b>26</b>
3.1	Introduction . . . . .	26
3.2	Partie électronique du prototype . . . . .	26
3.2.1	Matériel utilisé . . . . .	26
3.2.1.1	Arduino UNO R3 . . . . .	27
3.2.1.2	Carte Wi-Fi ESP-32S . . . . .	27
3.2.1.3	Capteur de température et d'humidité . . . . .	28
3.2.1.4	Anémomètre . . . . .	29
3.2.1.5	Afficheur LCD . . . . .	32
3.2.1.6	Module I2C . . . . .	32
3.2.1.7	Plaque d'essai . . . . .	33
3.2.1.8	Câbles . . . . .	33
3.2.2	Logiciels utilisés . . . . .	34
3.2.2.1	Arduino IDE . . . . .	34
3.2.2.2	Proteus . . . . .	35
3.2.2.3	ThingSpeak . . . . .	36
3.2.3	Simulation du prototype . . . . .	36
3.2.3.1	Simulation du capteur DHT11 avec l'afficheur LCD . . . . .	36

3.2.3.2	Simulation du capteur de vitesse de vent (Moteur DC)	37
3.2.3.3	Simulation de DHT11 et Anémomètre avec l'afficheur I2C-LCD(16x2) . . . . .	38
3.2.3.4	Simulation de prototype avec carte Wi-Fi ESP-32S . . . . .	38
3.3	Partie Machine Learning du prototype . . . . .	39
3.3.1	Outils . . . . .	39
3.3.1.1	Python et Jupyter Notebook . . . . .	39
3.3.1.2	Bibliothèques . . . . .	40
3.3.2	Méthodologie et préparation des données . . . . .	42
3.3.2.1	Création du dataset . . . . .	42
3.3.2.2	Nettoyage des données . . . . .	45
3.3.2.3	Visualisation des données . . . . .	46
3.3.2.4	Partitionnement des données . . . . .	48
3.3.3	Modèles de classification . . . . .	49
3.3.3.1	Arbres de décision . . . . .	49
3.3.3.2	Forêt aléatoire . . . . .	52
3.3.3.3	Classifieur Bayésien Naïf . . . . .	54
3.3.3.4	k plus proches voisins . . . . .	55
3.4	Design du prototype en 3D . . . . .	57
3.5	Réalisation du prototype . . . . .	58
3.5.1	Schéma fonctionnel du système proposé . . . . .	58
3.5.2	Réalisation réelle . . . . .	58
3.5.3	Connexion à ThingSpeak . . . . .	60
3.5.4	Communication Arduino-Python et Alertes . . . . .	61
3.6	Conclusion . . . . .	63
<b>CONCLUSION GÉNÉRALE ET PERSPECTIVES</b>		<b>65</b>
<b>Annexe</b>		<b>70</b>

---

## TABLE DES FIGURES

1.1	Carte des incendies actifs dans le monde . . . . .	3
1.2	Causes de la production d'une inflammation. . . . .	4
1.3	Superficie totale des incendies brûlés cartographiés par pays en 2021. . . . .	4
1.4	Evolution de la zone des feux de forêt de 1960 à 2020 au Maroc. . . . .	5
1.5	Schéma fonctionnel du système de détection des incendies de forêt basé sur l'IoT. . . . .	6
1.6	Schéma fonctionnel du système de détection des incendies de forêt basé sur l'IoT et API Cloud. . . . .	7
1.7	Système d'alerte précoce des incendies de forêt basé sur UAV. . . . .	8
1.8	Besoins fonctionnels. . . . .	10
1.9	Diagramme de Gantt du projet. . . . .	12
1.10	Modèle en V du cycle de vie de notre prototype. . . . .	14
2.1	Internet des objets. . . . .	17
2.2	Types de l'AI. . . . .	19
2.3	Quelques applications de l'AI. . . . .	19
2.4	Différence entre AI et ML et DL. . . . .	20
2.5	4 notions clefs du ML. . . . .	21
2.6	Exemple du ML supervisé. . . . .	22
2.7	Différence entre la classification et la régression. . . . .	23
2.8	Exemple du ML non supervisé. . . . .	23
2.9	Schéma explicatif du fonctionnement du ML par renforcement. . . . .	24
2.10	Réseau de neurones. . . . .	25
3.1	Carte Arduino UNO R3. . . . .	27

3.2	Carte Wi-Fi ESP-32S.	28
3.3	Capteur de température et d'humidité (DHT11).	28
3.4	Anémomètre.	29
3.5	Anémomètre fabriqué.	30
3.6	Graphe représentant la relation entre la vitesse du vent et la tension.	31
3.7	Afficheur LCD.	32
3.8	Module I2C.	33
3.9	Plaque d'essai.	33
3.10	Câble mâle-mâle.	34
3.11	Câble mâle-femelle.	34
3.12	Câble USB.	34
3.13	Câble Micro-USB.	34
3.14	Arduino IDE.	35
3.15	Interface du logiciel Proteus.	35
3.16	Logo de la plateforme ThingSpeak.	36
3.17	Simulation du capteur DHT11 avec l'afficheur LCD-I2C.	37
3.18	Simulation de l'anémomètre.	37
3.19	Simulation de DHT11 et anémomètre avec l'afficheur LCD-I2C.	38
3.20	Simulation de prototype avec la carte Wi-Fi ESP-32S.	39
3.21	Logo de Python.	40
3.22	Logo de Jupyter.	40
3.23	Logo du package Numpy.	40
3.24	Importation de la bibliothèque Numpy sur Jupyter.	40
3.25	Logo du package Pandas.	41
3.26	Importation de la bibliothèque Pandas sur Jupyter.	41
3.27	Logo du package matplotlib.	41
3.28	Importation de la bibliothèque matplotlib sur Jupyter.	41
3.29	Logo du package SKlearn.	42
3.30	Importation de la bibliothèque Sklearn sur Jupyter.	42
3.31	Lignes du code pour créer le dataset.	43
3.32	Base de données sous format CSV.	44
3.33	Extrait de la base de données sous format Excel.	44
3.34	Lignes du code pour l'importation et le partitionnement de la base de données sur Jupyter.	45
3.35	Lignes du code pour nettoyer le dataset.	46
3.36	Lignes du code pour visualiser les données	46
3.37	Résultat de la visualisation.	47

3.38	Code de dominance et son résultat graphique.	48
3.39	Importation des bibliothèques nécessaires pour le partitionnement des données.	49
3.40	Exemple des arbres de décision en ML.	49
3.41	Importation des bibliothèques nécessaires pour le modèle des arbres de décision.	51
3.42	Courbe des scores du modèle des arbres de décision.	51
3.43	Lignes du code du modèle des arbres de décision.	52
3.44	Schéma explicatif du modèle de la forêt aléatoire.	52
3.45	Importation des bibliothèques nécessaires pour le modèle de la forêt aléatoire.	53
3.46	Courbe des scores du modèle de la forêt aléatoire.	53
3.47	Lignes du code du modèle de la forêt aléatoire.	54
3.48	Importation de la bibliothèque du modèle du Bayésien Naïf.	54
3.49	Entrainement du modèle du Bayésien Naïf.	55
3.50	Scores du modèle du Bayésien Naïf.	55
3.51	Importation des bibliothèques du modèle KNN.	55
3.52	Courbe des scores du modèle KNN.	56
3.53	Lignes du code du modèle KNN.	56
3.54	Design du prototype en 3D.	57
3.55	Schéma fonctionnel du système proposé.	58
3.56	Prototype de la forêt en réalité.	59
3.57	Prototype de la forêt avec la station de surveillance en réalité.	59
3.58	Variation de la température et d'humidité.	60
3.59	Augmentation de température et diminution d'humidité.	60
3.60	Visualisation des données sur ThingSpeak.	61
3.61	Lignes du code pour afficher les données du moniteur série avec leurs prévisions.	62
3.62	Données en temps réel avec leurs classes.	62
3.63	Lignes du code de l'e-mail.	63
3.64	E-mail envoyé.	63

---

## LISTE DES TABLEAUX

1.1	Avantages et inconvénients des travaux connexes. . . . .	9
1.2	Tableau des tâches. . . . .	11
2.1	4 couches de l'internet des objets. . . . .	18
3.1	Matériel utilisé dans ce prototype. . . . .	26
3.2	Mesures de la vitesse de vent et la différence de tension . . . . .	31
3.3	Comparaison des scores des différents modèles . . . . .	57

---

## LISTE DES ABRÉVIATIONS

**IOT** : Internet Of Things.

**AI** : Artificial Intelligence.

**EFFIS** : European Forest Fire Information System.

**GPRS** : General Packet Radio Service.

**GSM** : Global System for Mobile Communications.

**SMS** : Short Message Service.

**HTTP** : HyperText Transfer Protocol.

**API** : Application Programming Interface.

**UAV** : Unmanned Aerial Vehicle.

**GPS** : Global Positioning System.

**IMU** : Inertial Measurement Unit.

**RFID** : Radio Frequency Identification.

**DHT11** : Digital Humidity and Temperature.

**ML** : Machine Learning.

**DL** : Deep Learning.

**3D** : three Dimensions or three-Dimensional.

**E/S** : Entrées/sorties.

**USB** : Universal Serial Bus.

**ICSP** : In Circuit Serial Programming.

**IDE** : Integrated Development Environment.

**Wi-Fi** : Wireless Fidelity.

**SoC** : System on a Chip.

**VCC** : Voltage Common Collector.

**GND** : Ground.

**DC** : Direct Current.

**LCD** : Liquid Crystal Display.

**I2C** : Inter-Integrated Circuit.

**SCL** : Serial Clock.

**SDA** : Serial Data.

**MAC** : Macintosh.

**PCB** : Printed Circuit Board.

**Rx** : Receive.

**Tx** : Transmit.

**COM** : Communication.

**CSV** : Comma-Separated Values.

**KNN** : KNeighboursClassifier.

---

# INTRODUCTION GÉNÉRALE

Chaque année, de 3 à 10 millions hectares sont brûlés, de 60 à 80 mille incendies sont détectés [1]. Cela est dû aux activités humaines en première classe, en outre des conditions météorologiques défavorables. Ces incendies impactent notre planète au niveau écologique, économique et santé public.

L'Australie détecte le nombre le plus élevé des feux à travers le monde. En Brésil, les incendies dévastent surtout dans les forêts tropicales d'Amazonie à cause des activités illégales, ainsi que le changement climatique qui était développé ces dernières années joue un rôle crucial dans l'inflammation des feux de forêt de Californie en Etats-Unis. En plus, les pays de la Méditerranée ont été attaqués par une séquence d'incendies terribles en 2022, particulièrement le Maroc (27.000 ha)[2].

Notre proposition est un outil efficace pour prévoir les incendies de forêt avant leur occurrence, en se basant sur les conditions météorologiques. C'est un système intelligent qui combine l'IoT(Internet Of Things) et l'AI(Artificial Intelligence). Plus précisément, le ML(Machine Learning) qui permettra la prévision des incendies de forêt en temps réel à l'aide des capteurs intelligents de température, humidité et de vitesse du vent.

Le présent rapport est organisé en 3 chapitres. Le premier est consacré à la détermination du cahier des charges, comprenant la problématique, les besoins fonctionnels, le plan de travail, etc. Le deuxième chapitre a pour objectif de définir les éléments clés de notre projet tels que l'IoT, l'AI et le ML. Le troisième chapitre est dédié à la conception du projet, tant au niveau de l'électronique (matériel, logiciels, simulation, etc.) que de l'informatique (base de données, modèle d'apprentissage). Il présente également la réalisation effective de notre projet, incluant les résultats du système tels que la visualisation des variations des données (température, humidité et vitesse du vent) et l'alerte de prévision d'incendie. Enfin, ce travail se conclut par une conclusion générale et des perspectives.

---

---

# CHAPITRE 1

---

## PRÉSENTATION ET CADRAGE DU PROJET (CAHIER DE CHARGE)

### 1.1 Introduction

**A**u cours de ce premier chapitre, nous allons élaborer une présentation générale du sujet, du contexte, de la problématique à résoudre et des objectifs du projet, ainsi que mettre en évidence les projets qui ont déjà traité le même sujet. Nous présenterons également notre solution proposée, les fonctionnalités attendues et le déroulement du projet.

### 1.2 Présentation du sujet

Chaque année, le nombre d'incendies survenus dans les forêts est en augmentation, notamment ces dernières années, ce qui entraîne de graves conséquences sur l'environnement, l'économie et la société. Face à cette situation, il devient crucial de mettre en place un système intelligent de prévision des incendies, permettant d'anticiper leur survenue ou de les maîtriser rapidement afin de réduire les dégâts et de protéger la population.



FIGURE 1.1 – Carte des incendies actifs dans le monde.

## 1.3 Contexte du projet

Afin d'obtenir le diplôme de la licence en électronique électrotechnique et automatique, au sein de la faculté des sciences et techniques d'Al-Hoceima, dans le cadre du projet de fin d'études, notre objectif est de réaliser un prototype d'un système de prévision des incendies de forêt en se basant sur IOT et AI, permettant d'anticiper les incendies avant leur occurrence pour minimiser les dommages provoqués par ces incendies.

## 1.4 Problématique

Depuis des années, le monde entier est impacté par les feux de forêt. Ce sont des incendies à grande échelle qui peuvent endommager considérablement les zones rurales et urbaines. Ils se propagent rapidement et ont la capacité de changer de direction. De tels incendies peuvent être provoqués par des causes naturelles telles que le réchauffement climatique et la foudre, ou par la négligence d'un individu [3].

La production d'une inflammation ou une combustion est liée aux trois facteurs suivants :

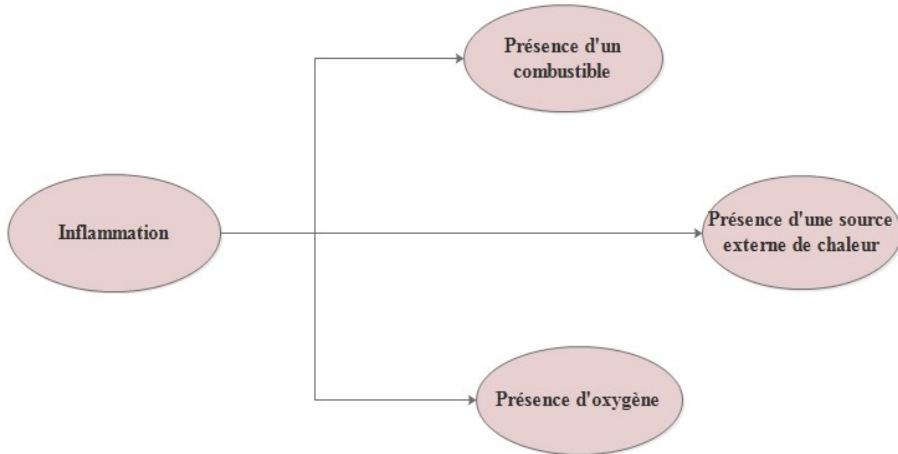


FIGURE 1.2 – Causes de la production d'une inflammation.

#### Dans le monde :

Dans ce cadre, l'EFFIS (European Forest Fire Information System) a réalisé une étude approfondie sur les incendies de forêt dans tous les pays du monde en 2021. Voici un histogramme représentant la superficie totale des incendies par pays en 2021 [4] :

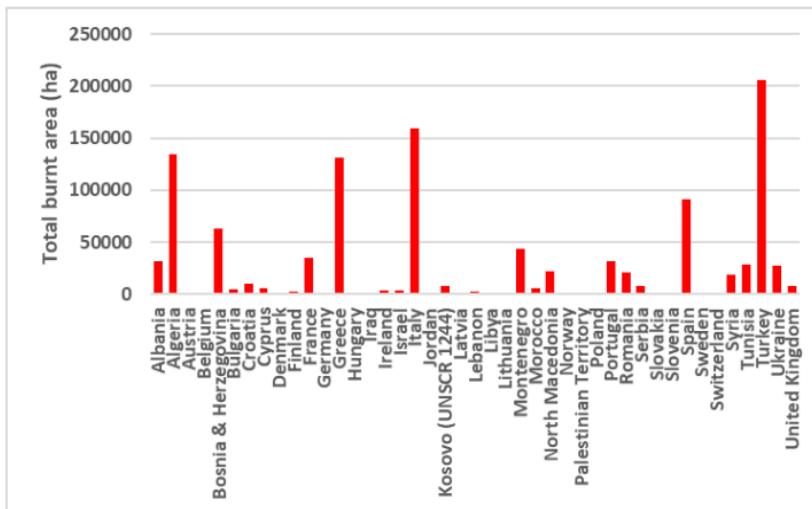


FIGURE 1.3 – Superficie totale des incendies brûlés cartographiés par pays en 2021.

#### Au Maroc :

En raison de la forte inflammabilité des espèces forestières tout au long de l'été, les zones forestières du Maroc connaissent un risque récurrent d'incendie, tout comme dans

les pays méditerranéens. Ce risque a des effets socio-économiques et environnementaux. La courbe ci-dessous représente l'évolution de la zone des feux de forêt de 1960 à 2020 [5] :

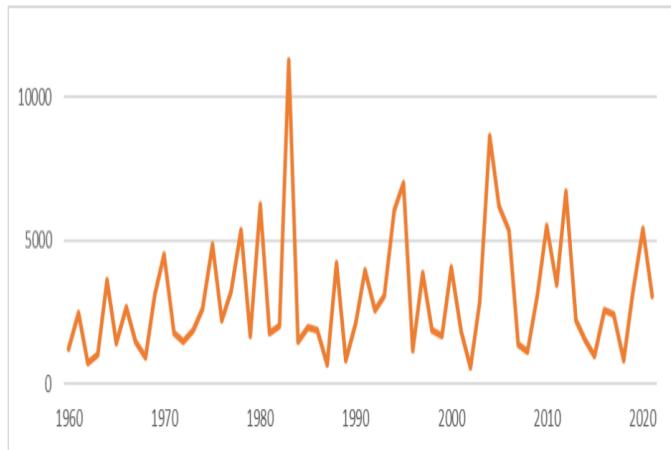


FIGURE 1.4 – Evolution de la zone des feux de forêt de 1960 à 2020 au Maroc.

## 1.5 Objectifs

Développer une solution avancée de prévision des incendies de forêt qui repose sur une technologie intelligente, aide à une amélioration de la gestion des incendies. Notre projet a pour objectif ultime de réduire les répercussions néfastes des incendies forestiers sur l'environnement, l'économie et les communautés.

## 1.6 Etude de l'existant

Dernièrement tous les phénomènes naturels ont subi une technologie qui permet de prévoir le phénomène pour minimiser la destruction environnementale qu'il puisse aboutir, notamment les incendies de forêt car ils sont les plus fréquentés par l'être humain.

Il y a beaucoup de systèmes qui ont implémenté un système intelligent de prévision des incendies de forêt. Par exemple dans [6], à l'aide de la science des données et du ML en utilisant Python comme langage de programmation, ils ont créé un modèle qui englobe l'ensemble de données sur les incendies détectés et détecte les incendies futurs en fonction d'aspects météorologiques sélectionnés.

De plus, dans [7], ils utilisent un ensemble de données contenant des informations telles

que la température, l'humidité, le vent et d'autres variables météorologiques. Ces données sont prétraitées et utilisées pour entraîner un algorithme de classification qui peut prévoir si une zone donnée est susceptible de subir un incendie de forêt, et afficher les données sur une application "Forest Fire Prediction App" avec la probabilité d'incendie. De même, dans [8], le feu de forêt est prévu en fonction de la température et d'humidité. Cet ensemble de données est préparé à l'aide des incendies de forêt qui ont eu lieu précédemment, et la mesure se fait durant ces incendies. Et le modèle du ML a été formé à l'aide du classificateur de régression logistique avec une précision de 91%.

Egalement dans [9], l'objectif du projet est de construire un système de détection d'incendie de forêt en utilisant l'IoT qui détecte l'incendie et envoie une alerte d'urgence à l'Autorité via l'IoT et un module compact SIM800L qui permet la transmission GPRS (General Packet Radio Service), envoyer/recevoir des SMS (Short Message Service) et passer des appels vocaux.

La figure ci-dessous représente le schéma fonctionnel de ce projet :

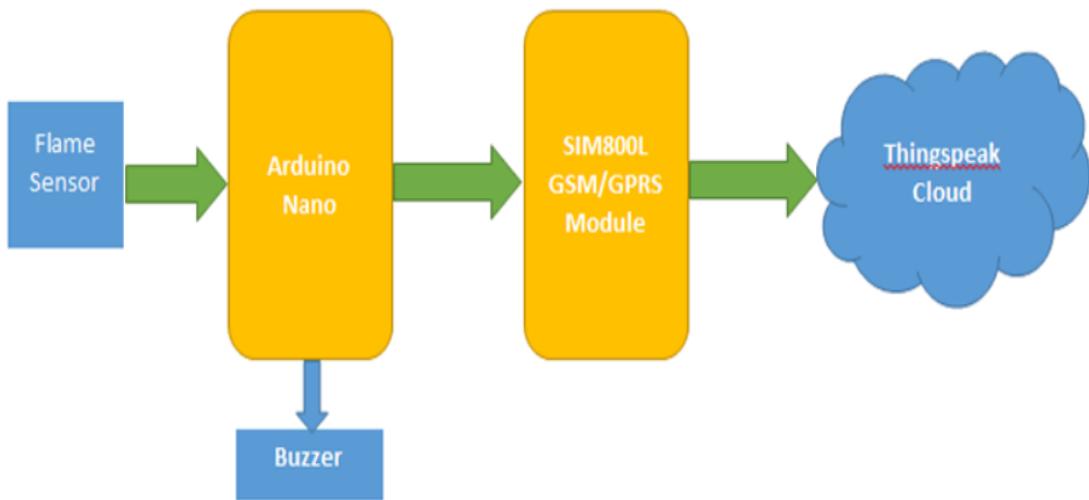


FIGURE 1.5 – Schéma fonctionnel du système de détection des incendies de forêt basé sur l'IoT.

Ainsi que dans [10], le système est développé pour surveiller les incendies de forêt via l'IoT, l'Arduino, et le cloud open source Thingspeak, utilisant le protocole HTTP (HyperText Transfer Protocol) pour transférer les informations du capteur vers le cloud. Et parmi les capteurs utilisés il y a le capteur de la température, l'humidité, l'intensité lumineuse, les ultrasons, l'accéléromètre. Ils supervisent les valeurs du capteur et envoient un mail à l'utilisateur concerné en utilisant l'API (Application Programming Interface) cloud. Leur système satisfait les paramètres de surveillance via le cloud.

La figure ci-dessous représente le schéma fonctionnel de ce projet :

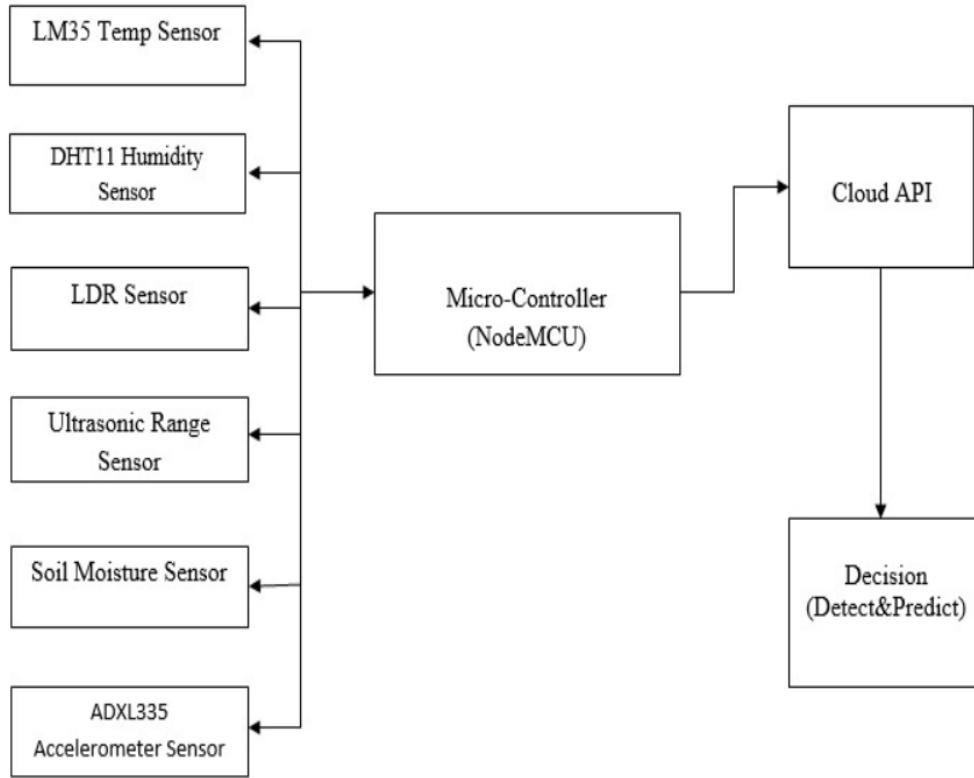


FIGURE 1.6 – Schéma fonctionnel du système de détection des incendies de forêt basé sur l'IoT et API Cloud.

En outre, dans [11], il s'agit d'un UAV (Unmanned Aerial Vehicle) qui est un véhicule autonome aérien relié à une station de contrôle équipée d'une caméra, d'un GPS (Global Positioning System), d'une IMU (Inertial Measurement Unit) et d'un capteur thermique. Il vole à une hauteur spécifique pour prendre une grande image de la forêt. En se déplaçant vers les zones chaudes, il capture des images détaillées à traiter et analyser pour prévoir les feux de forêt et leur propagation. Lorsqu'une alarme incendie se déclenche, les pompiers sont notifiés via une connexion Internet. Cette étude se concentre exclusivement sur le processus de photogrammétrie pour les données d'images. La figure ci-joint est une illustration du projet :

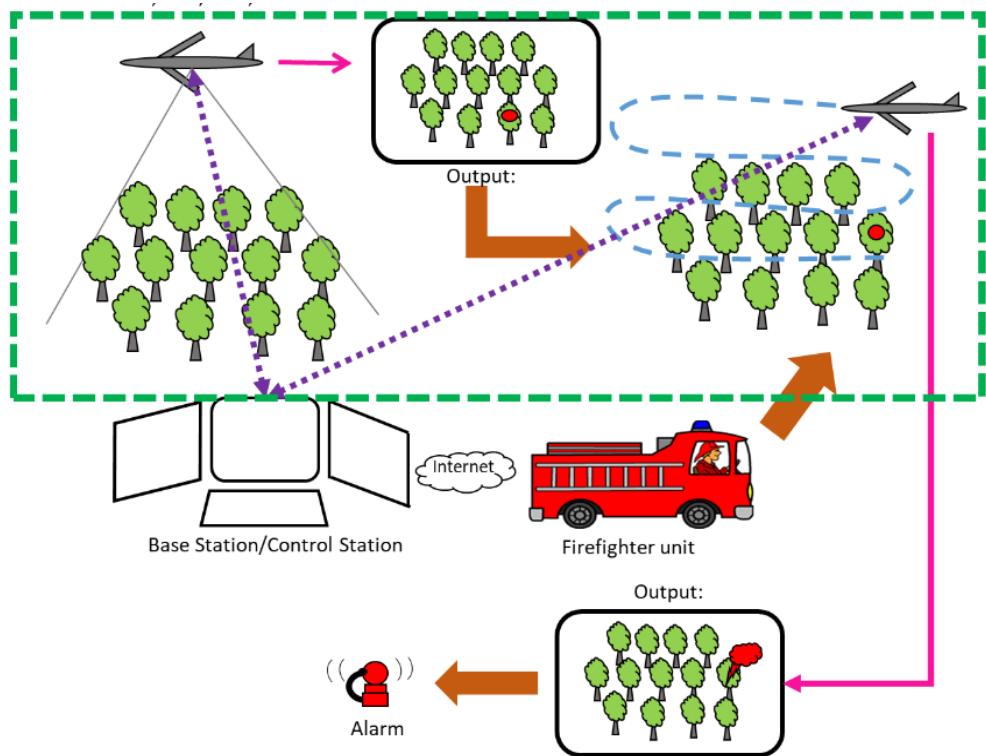


FIGURE 1.7 – Système d'alerte précoce des incendies de forêt basé sur UAV.

Finalement, dans [12], ils ont proposé une approche incorporée déterministe et probabiliste pour prévoir les dangers d'incendie de forêt dus à l'activité de la foudre en terrain montagneux. À la suite de leur étude, un critère probabiliste d'évaluation de la survenue d'un feu de forêt causé par l'impact de la foudre a été élaboré. La modélisation de scénarios et l'analyse paramétrique ont permis d'établir que la probabilité calculée d'occurrence d'un feu de forêt reflète adéquatement le changement des données initiales. De plus, des résultats cohérents ont été obtenus pour les scénarios d'activité de foudre faible, modérée et élevée.

### Récapitulatif :

Voici ci-dessous un récapitulatif qui résume tous les projets abordés avec leurs avantages et inconvénients :

Projet	Avantages	Inconvénients
[6]	- Utilisation du ML : algorithmes de classification et régression - Visualisation des données	- Dépendance totale sur les données historiques - Précision de prévision relative
[7]	- Utilisation du ML : algorithmes de classification	- Précision de 53%
[8]	- Utilisation du ML : algorithmes de régression - Précision de 91%	- Les variables environnementales insuffisantes
[9]	- Utilisation de la technologie IoT qui permet la surveillance à distance	- Il faut assurer une source d'alimentation continue - Les fausses alertes en raison de facteurs externes - Le module GSM peut être absent dans certaines régions
[10]	- Surveillance en temps réel - Faible coût	- Dépendance à l'Internet - Besoin d'un entretien régulier
[11]	- Couverture rapide d'une grande zone	- Coût élevé
[12]	- Surveillance en temps réel	- Complexité technique

TABLE 1.1 – Avantages et inconvénients des travaux connexes.

## 1.7 Produit du projet (Solution Proposée)

En vue des problèmes que peuvent causer les incendies de forêt, nous proposons la réalisation d'un prototype intelligent qui permet la prévision des incendies de forêt, en utilisant l'IOT qui sont des capteurs intelligents de température, d'humidité, et vitesse du vent, et les modèles du ML.

## 1.8 Besoins fonctionnels

Afin d'apporter une valeur ajoutée, notre projet nécessite un ensemble de fonctionnalités permettant de résoudre les problèmes rencontrés. Nous présentons ci-dessous les fonctionnalités requises :

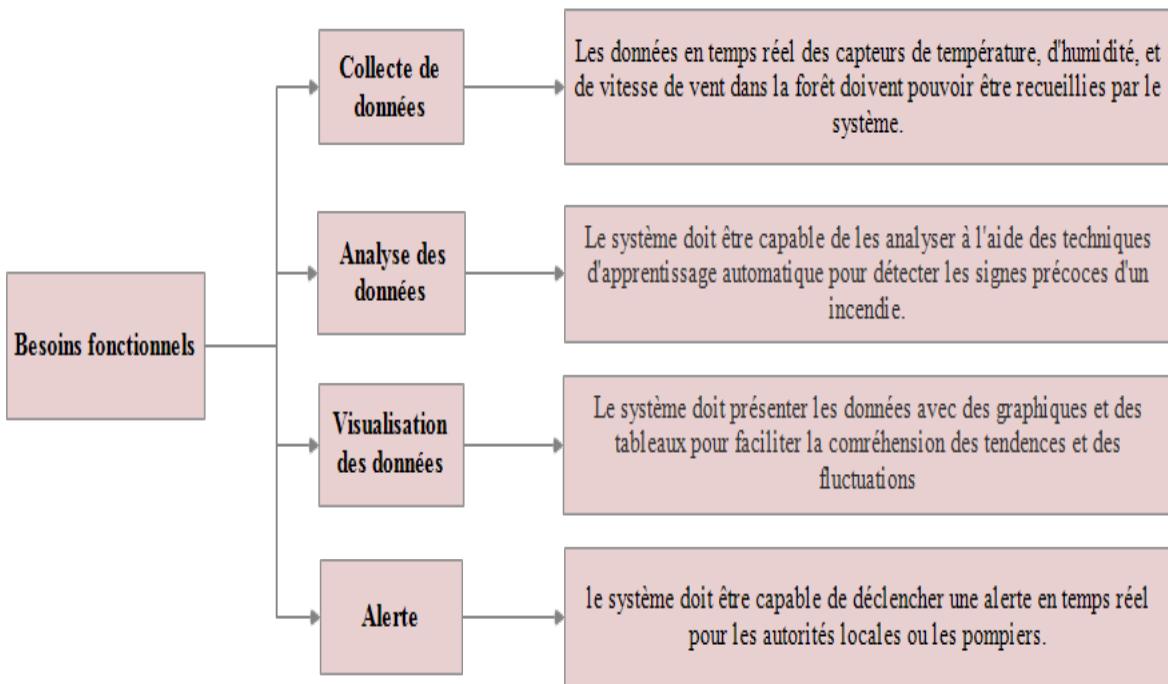


FIGURE 1.8 – Besoins fonctionnels.

## 1.9 Besoins non fonctionnels

Nous discutons des besoins qui ne sont pas directement liés aux fonctionnalités du produit, mais ils sont importés pour satisfaire :

**Facilité d'utilisation :** le système doit être facile à utiliser et à entretenir pour les utilisateurs.

**Coût :** le système doit être abordable pour les clients potentiels, en utilisant des composants peu coûteux et en évitant les coûts de développement élevés.

**Sécurité :** le système doit être sécurisé pour éviter toute intrusion ou piratage.

**Fiabilité :** le système doit être fiable et capable de fonctionner 24 heures sur 24, 7 jours sur 7.

**Précision :** le système doit être précis et être capable de détecter les signes d'un incendie dès qu'ils apparaissent.

**Évolutivité :** le système doit être conçu de manière évolutive pour permettre des mises à jour futures et des améliorations.

## 1.10 Déroulement du projet

### 1.10.1 Planification opérationnelle

Dans cette partie, on va planifier les étapes de notre projet, chaque étape avec sa date de départ, sa date finale et sa durée, en utilisant le tableau des tâches et par le diagramme de GANTT.

#### 1.10.1.1 Tableau des tâches

Tâche	Date de départ	Date de fin	Durée (jours)
Etude préalable	01/04/2023	07/04/2023	7
Cahier de charge	08/04/2023	17/04/2023	10
Conception du prototype	18/04/2023	26/04/2023	9
Simulation du prototype	27/04/2023	07/05/2023	11
Développement	08/05/2023	28/05/2023	21
Réalisation du prototype	12/05/2023	31/05/2023	20
Expérimentation du produit	01/06/2023	06/06/2023	6
Rédaction du rapport	08/06/2023	06/06/2023	60
Présentation	07/06/2023	13/06/2023	6

TABLE 1.2 – Tableau des tâches.

#### 1.10.1.2 Clarification des tâches

**Etude préalable :** C'est la première étape qu'on a fait envers le sujet, pour se familiariser avec les nouvelles notions qu'apporte le sujet.

**Cahier de charge :** C'est la partie préalable de la conception et la réalisation du prototype, qui encadre le projet dans tous les sens, et donne une idée générale sur la problématique, les objectifs, les projets existants, les besoins fonctionnels du projet, etc.

**Conception :** Dans cette phase, on a fait une architecture de notre projet, où on a cherché tous les composants électroniques nécessaires (disponibilité, qualité, prix, etc.) et les logiciels employés (formations).

**Simulation :** Dans cette partie, nous avons simulé tous les composants électroniques, pour savoir au préalable les problèmes de connexions qu'on puisse affronter pendant la réalisation du prototype.

**Développement :** C'est la partie où on a créé notre propre base de données, sur laquelle on a entraîné plusieurs algorithmes de classification pour trouver l'algorithme qui donne les meilleurs scores.

**Réalisation :** A ce point-là, nous avons connecté les composants électroniques avec le microcontrôleur Arduino Uno et la carte Wi-Fi (Wireless Fidelity) ESP-32S, ainsi qu'on a établi la communication entre l'Arduino et le programme de ML.

**Expérimentation du produit :** Après la réalisation du prototype, nous avons essayé de faire un grand nombre des tests pour avoir plus de confiance vers notre produit.

**Rédaction du rapport :** Pour respecter le dernier délai de remise du rapport, on a divisé les tâches entre le binôme, en plus la rédaction était au fur à mesure avec le développement du projet.

**Présentation :** Pour résumer le rapport, nous avons effectué une présentation Power Point afin de mettre en évidence les points primordiaux de notre projet.

#### 1.10.1.3 Diagramme de GANTT

Le diagramme de GANTT est un outil de gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. Il répertorie toutes les tâches à accomplir pour mener le projet à bien, et indique la date à laquelle ces tâches doivent être effectuées.

L'outil permet de créer différents diagrammes tels que les diagrammes de Gantt, les diagrammes de ressources et des réseaux PERT.

Ci-dessous on a réalisé le diagramme de GANTT de notre projet dans le logiciel GANTT project, pour le planning suivant :

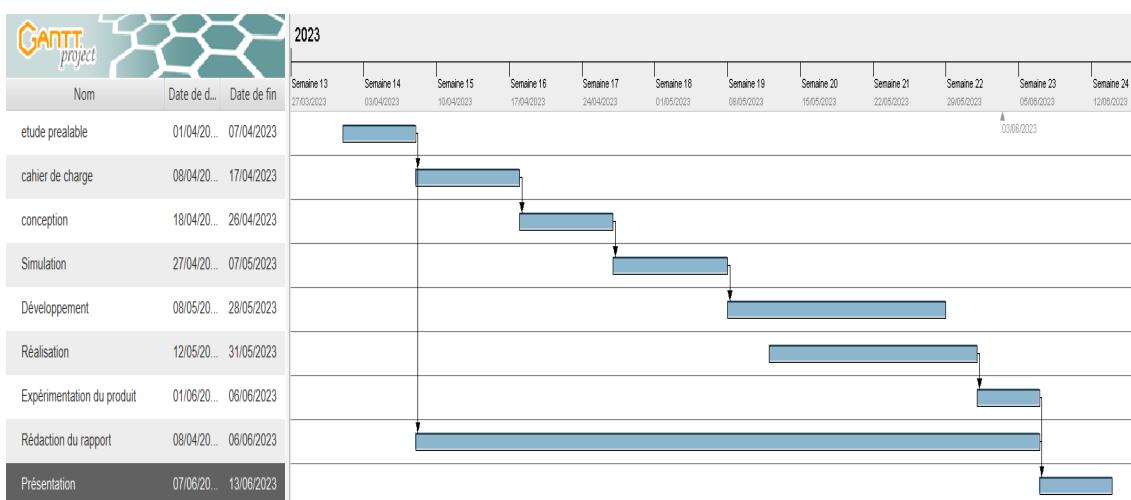


FIGURE 1.9 – Diagramme de Gantt du projet.

## 1.10.2 Processus de développement

### 1.10.2.1 Présentation du cycle de vie de développement

La présentation du cycle de vie de développement est une description systématique du processus de création d'un projet, depuis sa conception jusqu'à sa livraison finale. L'objectif principal de cette présentation est d'assurer la qualité et la fiabilité du projet en décrivant clairement le processus complet, y compris les tâches impliquées et les résultats souhaités.

Il existe plusieurs types [13] pour présenter un cycle de vie, à savoir le cycle de vie en V, le cycle de vie en cascade, le cycle de vie en spirale, le cycle de vie itératif, etc.

**Le modèle en V :** Cette méthode consiste à la création d'un modèle en forme de V, qui comprend deux branches principales : la branche de conception et la branche de validation.

**Justification du choix du cycle de vie en V :** Nous avons décidé d'utiliser le modèle en forme de V pour notre projet car :

- Il permet une meilleure gestion des risques : en identifiant et en évaluant les risques dès le début du projet, le cycle de vie permet de mettre en place des mesures pour les éviter ou les minimiser.
- Il facilite la communication entre les différentes parties prenantes : le cycle de vie en V est un modèle clair et facile à comprendre, ce qui permet aux différents acteurs du projet (développeurs, testeurs, clients, etc.) de communiquer plus facilement entre eux.
- Soutient la qualité du produit final - en intégrant les tests au début du projet, le cycle de vie en V permet de découvrir les problèmes plus tôt, facilite la correction des problèmes et améliore la qualité du produit final.
- Il permet de mieux contrôler les coûts et les délais : en identifiant les besoins et en planifiant les tests dès le début du projet, le cycle de vie en V permet de mieux anticiper les coûts et les délais, ce qui facilite la gestion du projet.

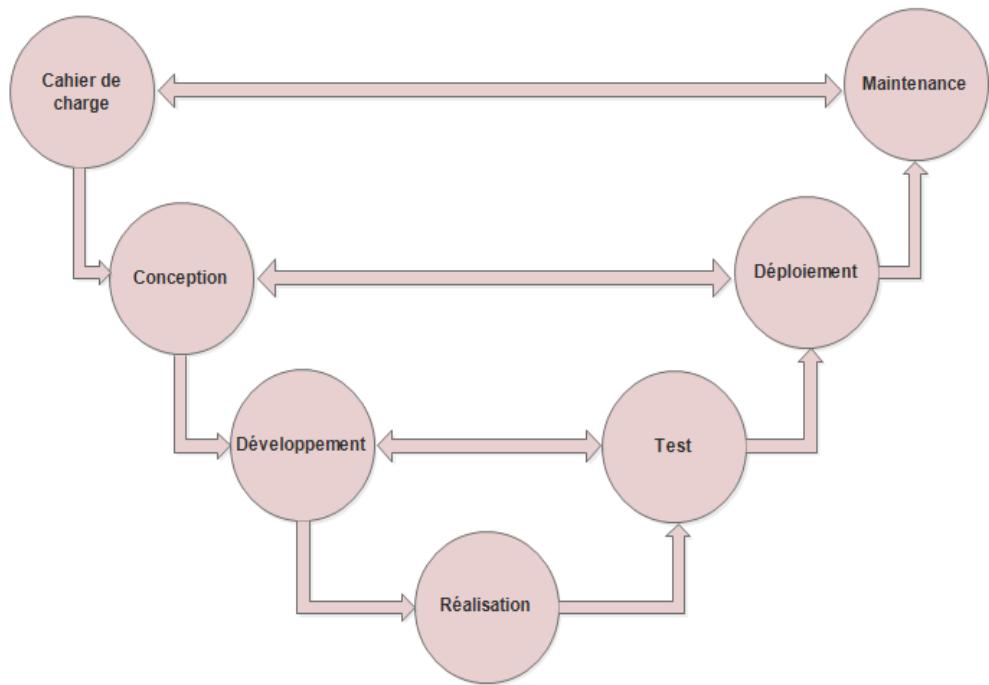


FIGURE 1.10 – Modèle en V du cycle de vie de notre prototype.

#### 1.10.2.2 Phases

- **Cahier de charge** : cette étape définit les objectifs du projet, les exigences utilisateur, fonctionnelles et techniques. On peut aussi dans cette étape déterminer les coûts du projet et les besoins en ressources.
- **Conception** : Au cours de cette phase, l'architecture du système est conçue conjointement avec les exigences relatives aux capteurs, aux dispositifs de collecte de données, aux réseaux de communication et aux algorithmes d'IA.
- **Développement** : La construction et l'essai des divers composants du système constituent cette étape. Il est nécessaire de programmer et d'intégrer des capteurs, des outils de collecte de données, des réseaux de communication et des algorithmes d'intelligence artificielle.
- **Test** : Pour s'assurer que chaque composant fonctionne correctement, le système est testé à cette étape dans le cadre de scénarios d'utilisation réels. Les capteurs, les dispositifs de collecte de données, les réseaux de communication et les algorithmes d'intelligence artificielle doivent tous être testés.
- **Déploiement** : Le système doit être installé sur le terrain et connecté aux réseaux de communication à cette étape. Les utilisateurs peuvent également recevoir une formation sur la façon d'utiliser la technologie.

- **Maintenance :** Cette étape consiste à assurer la maintenance régulière du système, y compris la surveillance des performances, la gestion des données et les mises à jour logicielles.

## 1.11 Conclusion

Dans ce chapitre, nous avons abordé plusieurs points essentiels pour entreprendre les premières pas vers la réalisation de notre prototype, qui est un système de prévision des incendies de forêt basé sur l'IoT et l'AI. Ces points incluent les objectifs du système, tels que la protection des forêts et de la population ainsi que la réduction des dégâts, l'étude des projets existants, la proposition du projet, ses besoins fonctionnels tels que la collecte et l'analyse des données, l'alerte et la visualisation des données, aussi que ses besoins non fonctionnels, tels que la fiabilité, la précision et la sécurité, etc. Nous avons également discuté la planification du travail et le cycle de vie de notre projet, qui est le cycle en V, allant de l'établissement du cahier de charge à la maintenance, avec une explication détaillée de chaque phase du cycle de vie.

---

---

# CHAPITRE 2

---

## GÉNÉRALITÉS SUR L'INTERNET DES OBJETS ET L'INTELLIGENCE ARTIFICIELLE

### 2.1 Introduction

DANS ce chapitre, nous plongeons au cœur des notions fondamentales qui constituent la base de notre projet. Nous allons définir en détail les concepts clés tels que l'IoT et l'AI, en mettant l'accent sur le ML, qui jouera un rôle central dans la réalisation de notre prototype. De plus, nous élargirons notre vision en explorant d'autres méthodes d'apprentissage qui complètent et enrichissent notre compréhension du sujet.

### 2.2 C'est quoi l'internet des objets ?

L'IOT a été désigné pour la première fois par Kevin Ashton en 1999, pour illustrer la puissance de la connexion de RFID (Radio Frequency Identification) utilisé dans les chaînes d'approvisionnement d'entreprise à l'Internet. Il s'agit d'une infrastructure essentielle pour soutenir la société de l'information, en permettant la connectivité et l'interfonctionnement des objets physiques ou virtuels grâce aux technologies de l'information et de la communication. Cette infrastructure permet d'accéder à des services avancés en interconnectant ces objets, en utilisant les technologies existantes ou émergentes [14]. Maintenant elle est incluse dans tous les secteurs par exemple le développement durable, la logistique, la domotique, la santé, etc.

Concernant ce projet de prévision des incendies des forêts, on va utiliser deux capteurs, le capteur de température et humidité DHT11 (Digital Humidity and Temperature) car une baisse d'humidité d'air et une augmentation de température peuvent rendre les forêts sujet à un incendie, et le capteur de vitesse de vent "anémomètre" est aussi un élément essentiel dans la prévision des incendies de forêt car il peut aider à estimer la rapidité de propagation des flammes.

Ces capteurs seront connectés à Internet afin de surveiller et collecter les données météorologiques, ce qui va nous permettre de créer un objet connecté.

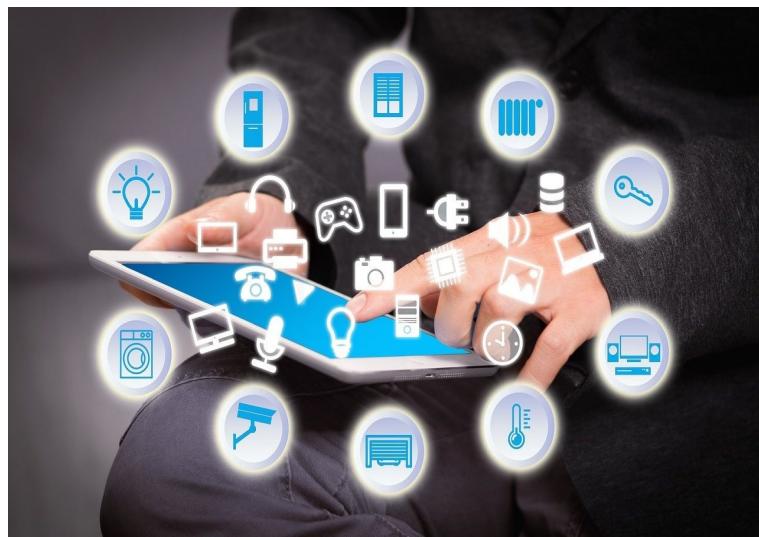


FIGURE 2.1 – Internet des objets.

### **Fonctionnement de l'internet des objets :**

Les capteurs sont placés dans des infrastructures physiques, puis ils transfèrent les données à l'aide d'un réseau sans fil à des plateformes IoT, soit pour les visualiser ou pour le contrôle. Cet ensemble créé un objet connecté.

On peut diviser l'IoT en 4 couches [15] :

Détection	Les capteurs intelligents
Réseautage	Réseau pour connecter les équipements entre eux
Service	Le logiciel médiateur qui permet de faire communiquer le logiciel et le matériel
Interface	Les plateformes qui présentent les applications aux utilisateurs

TABLE 2.1 – 4 couches de l'internet des objets.

## 2.3 Intelligence artificielle

En 1956 lors de la Conférence de Dartmouth le terme "Intelligence artificielle" a été inventé, où ils ont débattu la possibilité de créer des machines capables de penser comme des êtres humains.

On peut définir l'intelligence artificielle comme étant la capacité cognitive d'apprendre, d'en déduire des informations, de reconnaître des motifs répétitifs, et de résoudre des problèmes et de prendre des décisions. Cela ouvre des perspectives pour l'innovation et le développement de nouvelles solutions dans de nombreux domaines.

L'AI suscite néanmoins des interrogations éthiques et soulève des défis majeurs. Des préoccupations concernant la confidentialité des données, la transparence des décisions prises par les systèmes d'AI, l'impact sur l'emploi et la responsabilité en cas d'erreurs sont des sujets d'actualité qui requièrent une réflexion approfondie et une réglementation appropriée.

### 2.3.1 Types de l'Intelligence artificielle

Il existe 4 types d'intelligence artificielle [16] :

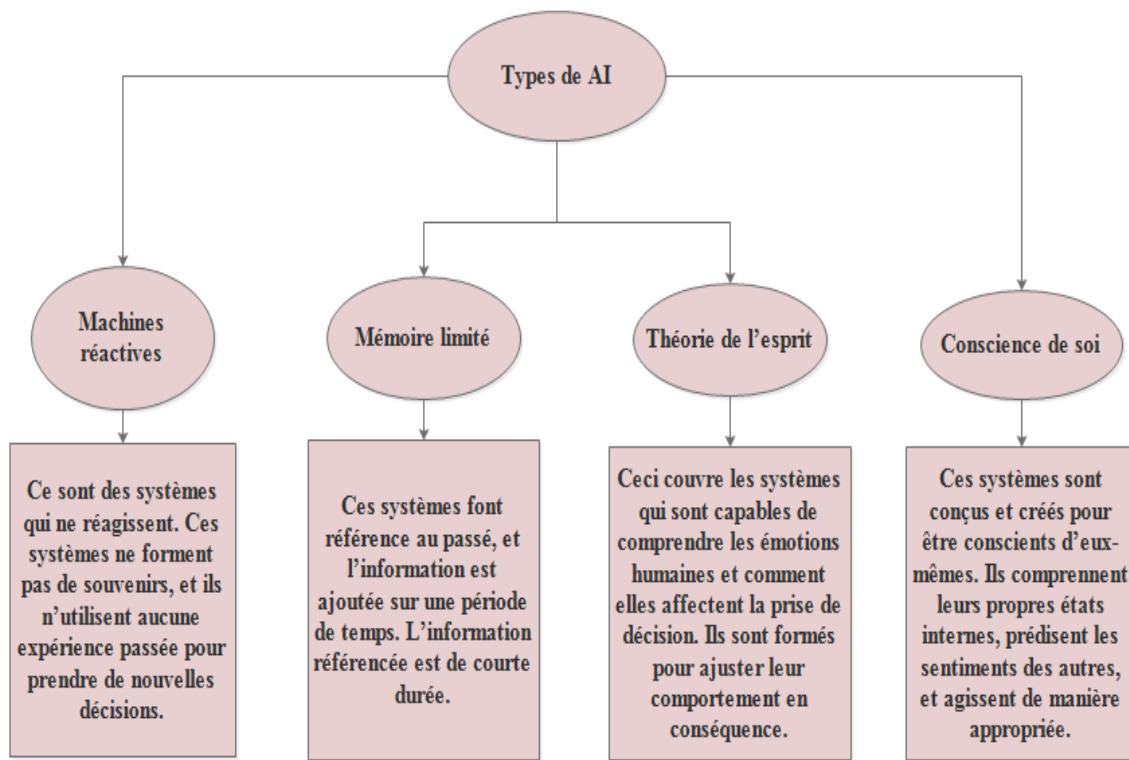


FIGURE 2.2 – Types de l'AI.

### 2.3.2 Applications de l'Intelligence artificielle

Il y a plusieurs applications d'AI sur le marché, voici quelques exemples :



FIGURE 2.3 – Quelques applications de l'AI.

**Google Assistant** : Il répond à toutes les questions de l'utilisateur et exécute ses ordres comme ouvrir la caméra, sonner à quelqu'un, etc.

**Google Lens** : Il permet l'identification des produits ou des endroits, la traduction d'un texte capturé par la caméra du smartphone, etc.

**Natureid** : Il permet l'identification de la plante et d'obtenir des informations détaillées sur la plante juste en prenant sa photo sur le smartphone.

**PicText** : Il permet de transformer une image à un texte.

### 2.3.3 Différence entre Intelligence artificielle et Machine Learning et Deep Learning

l'IA est un domaine plus large qui englobe le ML et le DL (Deep Learning). Le ML se concentre sur l'apprentissage à partir de données, tandis que le DL est une technique spécifique de ML utilisant des réseaux de neurones profonds pour apprendre des représentations complexes des données.

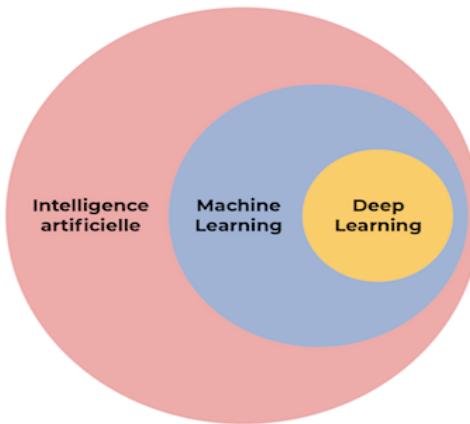


FIGURE 2.4 – Différence entre AI et ML et DL.

### 2.3.4 Machine Learning

La définition de ML d'après son inventeur Arthur Samuel en 1959, c'est la science qui permet aux ordinateurs d'apprendre sans être explicitement programmé.

Le ML est une discipline de l'informatique qui utilise des algorithmes informatiques et des analyses pour construire des modèles prévisionnels qui peuvent résoudre des problèmes.

Pour débuter un projet en ML, il faut savoir ces 4 notions clefs [17] :

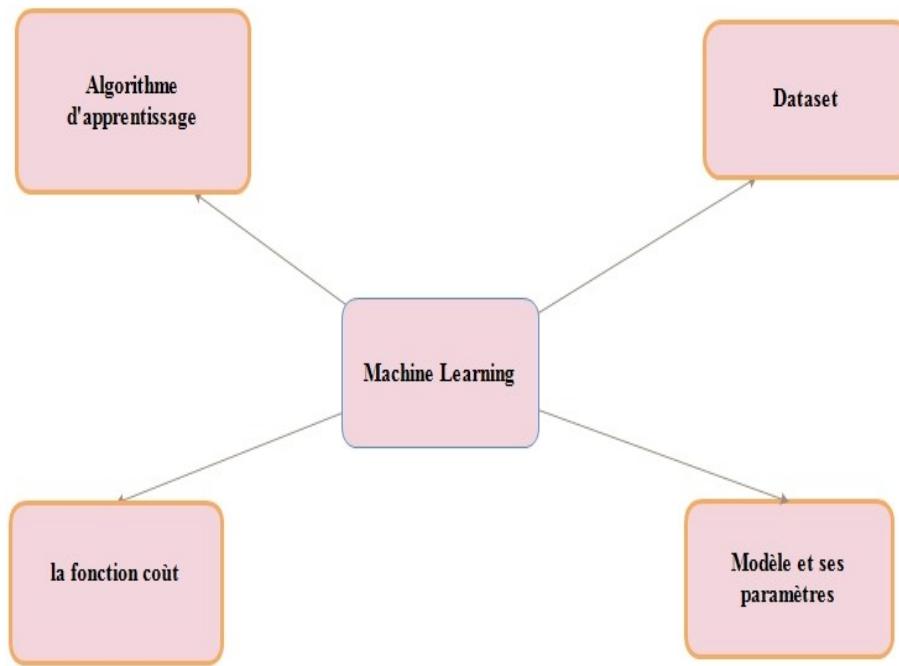


FIGURE 2.5 – 4 notions clefs du ML.

**Dataset** : C'est la base de données, où les  $x$  sont les entrées et les  $y$  sont les sorties.

**Modèle et ses paramètres** : le modèle est une fonction mathématique, qui est créé à partir des données qu'on a fourni, et les paramètres sont les coefficients de cette fonction mathématique.

**La fonction coût** : représente les erreurs qu'on obtient lorsque on teste notre modèle avec les données historiques.

**L'algorithme d'apprentissage** : on crée un algorithme pour laisser la machine à trouver les paramètres du modèle convenables au dataset afin de minimiser la fonction coût.

Il y a trois méthodes d'apprentissage qui sont toutes inspirées par l'être humain :

#### 2.3.4.1 Machine Learning supervisé

Le ML supervisé est une technique du ML dans laquelle un algorithme est formé sur un ensemble de données étiquetées, c'est-à-dire un ensemble de données dans lesquelles chaque échantillon est connecté à une étiquette ou une réponse préédéterminée, afin de prévoir la réponse pour des données inédites.

En résumé, le ML supervisé vise à créer un modèle prévisionnel qui peut généraliser les corrélations entre les caractères d'entrée et les étiquettes de sortie.

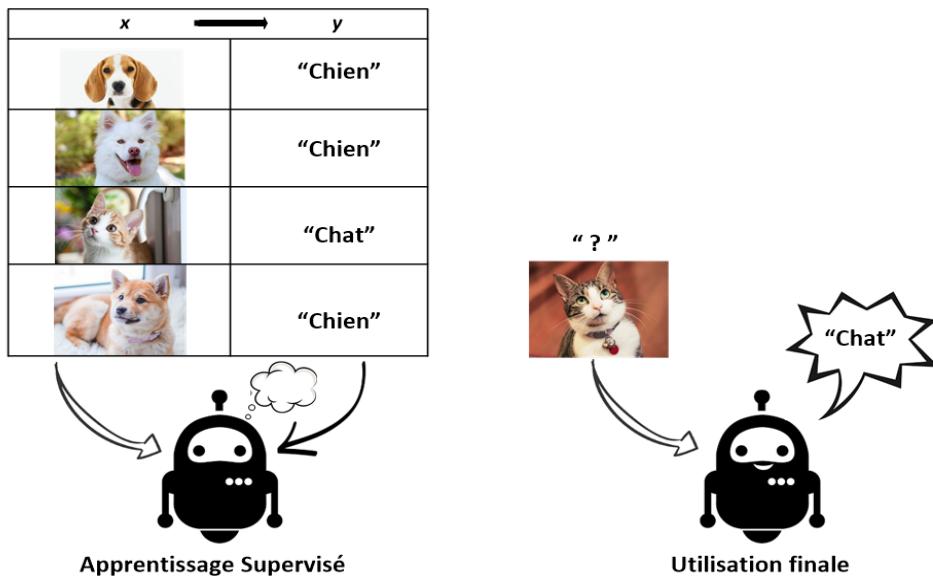


FIGURE 2.6 – Exemple du MI supervisé.

Il existe deux types d'applications de ML supervisé couramment utilisé pour résoudre des problèmes prévisionnels qui sont les algorithmes de régression et de classification [18] :

**Les algorithmes de régression :**

Les algorithmes de régression tentent à prévoir une valeur ou une quantité continue, l'objectif est de trouver une fonction mathématique qui peut prévoir les valeurs de la variable cible en fonction des valeurs de la variable d'entrée.

**Les algorithmes de classification :**

Les algorithmes de classification cherchent à prévoir une classe ou une catégorie, par exemple, on peut utiliser le ML supervisé pour classifier les emails comme spam ou non-spam.

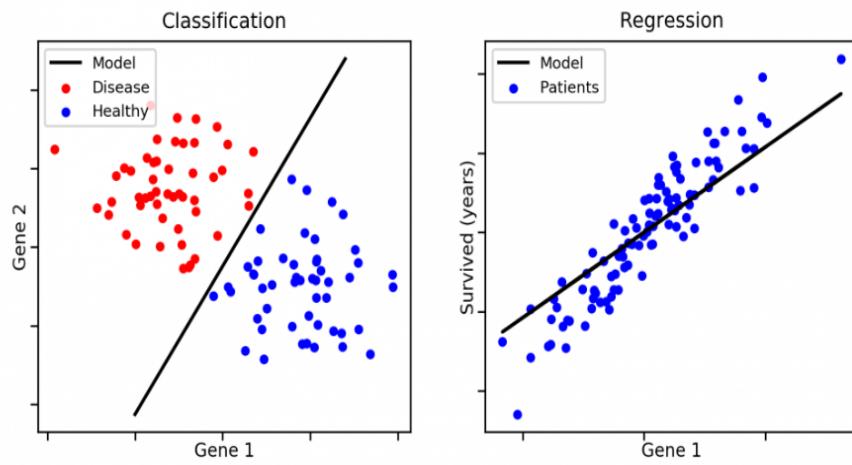


FIGURE 2.7 – Différence entre la classification et la régression.

#### 2.3.4.2 Machine Learning non supervisé

Le ML non supervisé est une méthode de ML qui consiste à construire un modèle à partir de données sans étiquetage ni résultats prédéterminés.

Contrairement au ML supervisé, le modèle doit découvrir indépendamment les structures et les modèles dans les données afin de prévoir les résultats.

Pour effectuer le ML non supervisé on utilise deux algorithmes primordiales qui sont les algorithmes de regroupement et d'association pour classer ces données afin de trouver une variable de sortie correspondante [18].

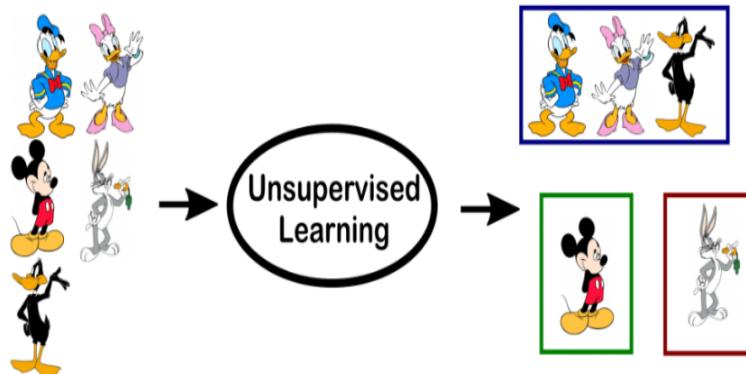


FIGURE 2.8 – Exemple du ML non supervisé.

### 2.3.4.3 Machine Learning par renforcement

Une méthode du ML appelée ML par renforcement s'occupe de créer des algorithmes qui peuvent apprendre à sélectionner le meilleur plan d'action dans des paramètres dynamiques compliqués. Avec la rétroaction sous forme d'incitations ou de pénalités, ces algorithmes apprennent par essais et erreurs.

Les applications telles que les jeux vidéo, pilotage des véhicules autonomes, contrôle des robots, utilisent fréquemment l'apprentissage du renforcement [19].

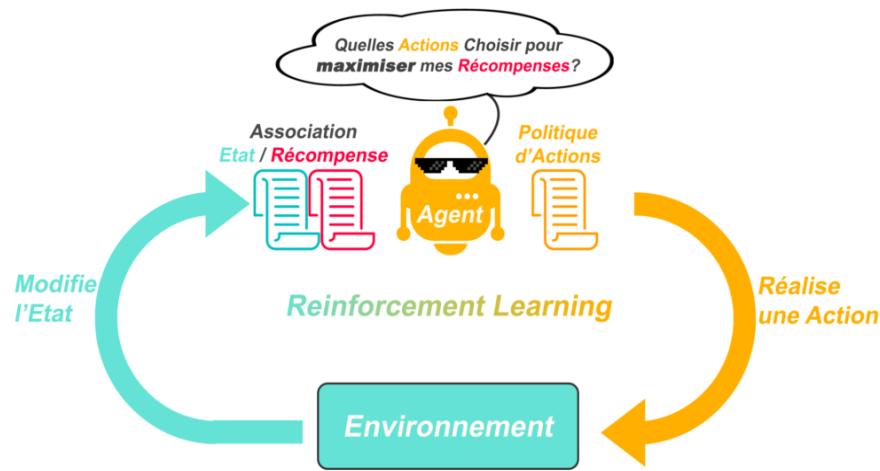


FIGURE 2.9 – Schéma explicatif du fonctionnement du ML par renforcement.

### 2.3.5 Deep Learning

Une branche du ML appelée DL se concentre sur la construction des systèmes qui peuvent apprendre, prévoir et prendre des décisions par eux-mêmes. Ce type d'AI utilise des algorithmes qui peuvent simuler le cerveau humain en exploitant un vaste réseau de neurones artificiels. Le traitement de grandes quantités de données afin d'améliorer les performances d'un algorithme et de créer un système intelligent et autonome est le principal avantage du DL [20].

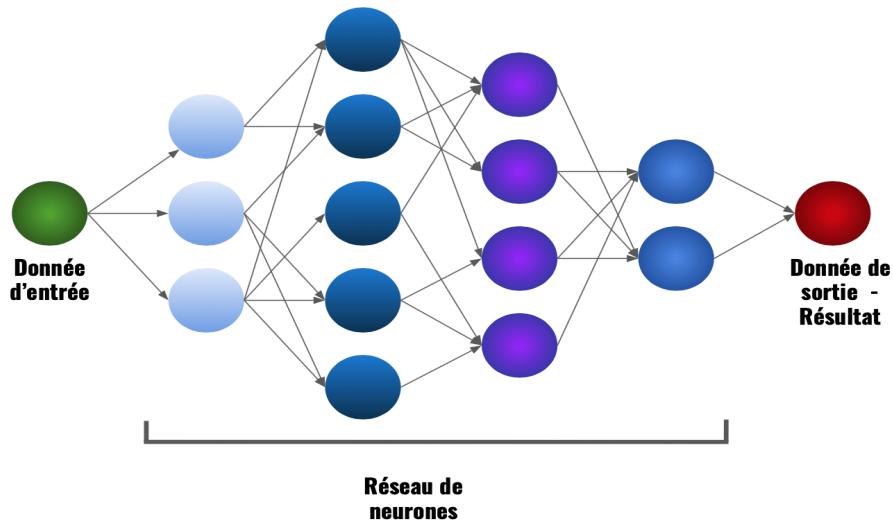


FIGURE 2.10 – Réseau de neurones.

## 2.4 Conclusion

Dans ce chapitre on a défini les principaux termes utilisés dans le cadre du projet qui sont l'IoT et l'AI, plus précisément le ML qui est utilisé souvent pour résoudre les problèmes prévisionnels.

L'IoT qui va nous servir à collecter les données pour les comparer avec les données déjà existantes dans une base de données (dataset), pour qu'au final à l'aide des algorithmes de classification supervisés, on aura la capacité de créer un modèle prévisionnel des incendies des forêts.

---

---

# CHAPITRE 3

---

## CONCEPTION ET RÉALISATION DU PROTOTYPE

### 3.1 Introduction

DANS ce chapitre on va entamer la partie électronique de notre prototype, telle que le matériel et les logiciels utilisés, et la simulation de chaque composant sur le logiciel PROTEUS. En plus, une partie de ML du prototype. Enfin, un design en 3D (three Dimensions or three-Dimensional) et la réalisation réelle de notre maquette.

### 3.2 Partie électronique du prototype

#### 3.2.1 Matériel utilisé

Matériels	Quantité
Arduino UNO R3	1
Carte ESP-32	1
DHT11	1
Moteur DC	1
Afficheur LCD-I2C(16x2)	1
Plaque d'essai	1
Les câbles	16

TABLE 3.1 – Matériel utilisé dans ce prototype.

### 3.2.1.1 Arduino UNO R3

L'Arduino Uno R3 est une carte électronique programmable basée sur un microcontrôleur ATMega328P. C'est l'une des cartes de microcontrôleur les plus appréciées de la famille Arduino et elle est conçue pour être simple à utiliser par les utilisateurs novices et experts. Les entrées comprennent 6 entrées analogiques, 14 broches d'E/S (Entrées/-Sorties) numériques, un port USB (Universal Serial Bus), une prise d'alimentation, un connecteur ICSP (In Circuit Serial Programming) et un bouton de réinitialisation. Elle peut être programmée avec le logiciel Arduino IDE (Integrated Development Environment) via un câble USB, ce qui permet de télécharger des programmes sur la carte et de la rendre autonome pour réaliser des projets électroniques [21].



FIGURE 3.1 – Carte Arduino UNO R3.

### 3.2.1.2 Carte Wi-Fi ESP-32S

Le module de développement ESP-32S est une carte électronique qui utilise le système sur puce ESP-32, créé par la société chinoise Espressif. Cette puce est fabriquée à l'aide d'une technologie avancée qui permet de combiner la connectivité Wi-Fi (Wireless Fidelity) et Bluetooth dans une seule puce compacte et économique en énergie. Le SoC (System on a Chip) ESP-32 est hautement intégré, nécessitant peu de composants externes, mais le module ESP-32S ajoute environ 20 composants externes pour offrir une solution complète pour les applications qui nécessitent Wi-Fi et Bluetooth. Cette carte est facile à employer et souvent utilisée pour prototyper des projets IoT [22].

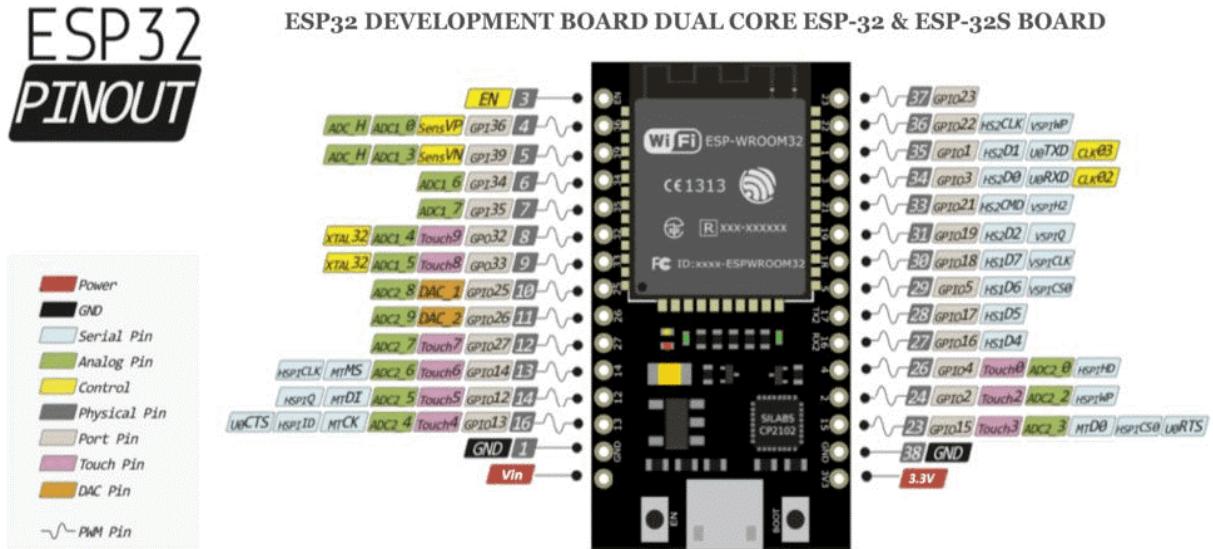


FIGURE 3.2 – Carte Wi-Fi ESP-32S.

### 3.2.1.3 Capteur de température et d'humidité

Lorsqu'on cherche à mesurer la température et l'humidité d'une manière simple, le capteur DHT11 est le modèle convenable à notre situation, il est couramment utilisé avec des microcontrôleurs comme Arduino, ainsi il peut être appliqué à une gamme de projets, y compris les systèmes de contrôle environnemental, les systèmes de surveillance météorologique et les systèmes de contrôle de la qualité, pour détecter la température et l'humidité de l'atmosphère [23].

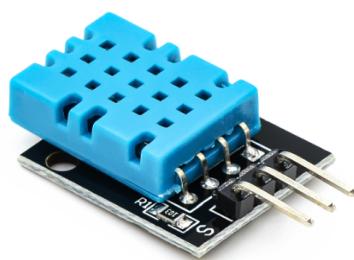


FIGURE 3.3 – Capteur de température et d'humidité (DHT11).

### **Caractéristiques de DHT11 :**

- Faible coût.
- La tension des E/S varie de 3,5 à 5,5 V.
- Le courant de fonctionnement maximal pour la conversion (lorsque des données sont demandées) est de 2,5 mA.
- Précision de 2 à 5 % pour des lectures d'humidité de 20 à 90 %.
- Bon pour lire des températures entre 0 et 50 °C ; précision 0,5 °C.
- Dimensions du corps : 12 mm, 15,5 mm et 5,5 mm.
- 2,4 g de poids.

[24]

### **Branchement :**

Concernant le branchement, il suffit de connecter la broche OUT du capteur DHT11 à un pin numérique de l'Arduino, après on relit la broche d'alimentation VCC (Voltage Common Collector) du capteur DHT11 à la broche 5V de l'Arduino, en fin on branche la broche GND (Ground) du capteur à la broche GND de l'Arduino.

#### **3.2.1.4 Anémomètre**

L'anémomètre est un appareil de mesure de la vitesse du vent. Il peut également donner des informations sur sa direction lorsqu'il est utilisé en conjonction avec une girouette. Bien que son application principale soit la météorologie, les données qu'il produit peuvent également être utilisées pour les voyages aériens, la navigation, etc[25].



FIGURE 3.4 – Anémomètre.

**Caractéristiques de l'anémomètre :**

- Entrée : 0,4 à 2 volts.
- Plage de test : 0,5 m/s à 50 m/s.
- Vitesse minimale du vent : 0,2 m/s.
- Résolution de 0,1 m/s.
- Vérité : pire des cas 1,0 m/s.
- Vitesse du vent maximale : 70 m/s.
- Informations sur le connecteur : la broche 1 est pour l'alimentation, la broche 2 est pour la masse, la broche 3 est pour le signal et la broche 4 n'est pas connectée.

Afin de réaliser ce prototype on a fabriqué un anémomètre construit à la base d'un moteur DC (Direct Current) et d'une turbine qui permet la rotation du moteur lorsque le vent souffle.

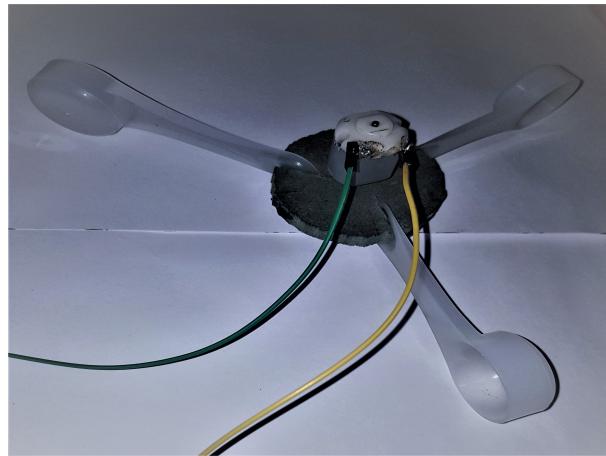


FIGURE 3.5 – Anémomètre fabriqué.

Un moteur à courant continu étant réversible et susceptible de se comporter selon deux modes soit en "moteur" soit en "génératrice". Cependant dans notre cas nous utilisons ce moteur en mode génératrice, alors que son fonctionnement est basé sur le principe de la conversion de l'énergie mécanique en énergie électrique. D'où La rotation de son rotor crée un champ magnétique qui traverse les bobines de l'induit, ce qu'induit une tension électrique et produit une courant électrique.

Dans [26], ils ont fait une expérience en utilisant leur voiture pour varier la vitesse, chaque 10km/h équivaut à une petite tension( en milli volts) affichée sur le multimètre. Durant cette étude ils ont trouvé ces mesures-là :

Vitesse du vent (km /h)	Différence de tension(mV)
20	82
30	125
40	190
50	250
60	320
70	390
80	430
90	480

TABLE 3.2 – Mesures de la vitesse de vent et la différence de tension

Alors il y a une relation presque linéaire entre la vitesse de vent et la différence de tension comme le montre la courbe ci-après :

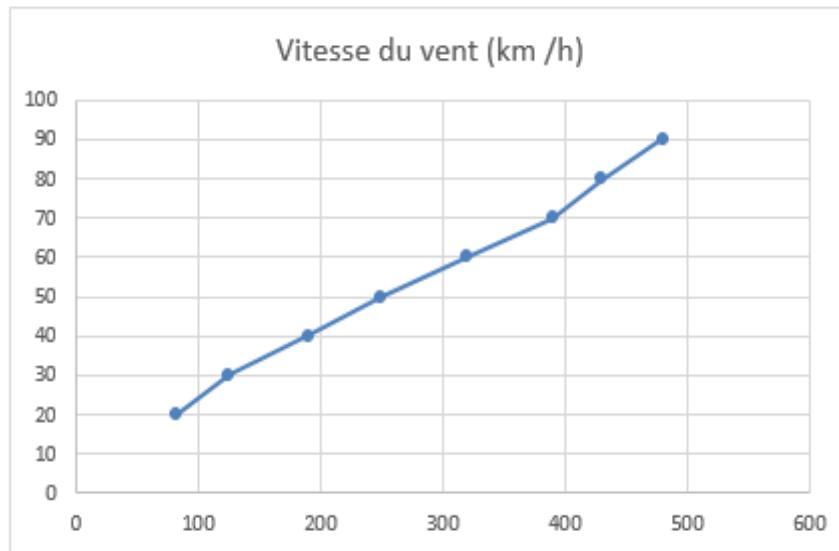


FIGURE 3.6 – Graphe représentant la relation entre la vitesse du vent et la tension.

On peut calculer la vitesse de vent à partir de la différence de potentiel qui va générer le moteur DC par la relation suivant :

$$\text{Vitesse\_du\_vent} = 0.190 \times \text{Différence\_de\_tension} \quad (3.1)$$

**NB :**

- Le facteur 0.190 est la moyenne de toutes les pentes de la courbe obtenue.
- Les valeurs de tension doivent être en millivolts.

### 3.2.1.5 Afficheur LCD

Les écrans à cristaux liquides, sont des modules compacts qui ne nécessitent que quelques composants supplémentaires pour fonctionner correctement. Ils consomment peu (entre 1 et 5mA). Il existe plusieurs afficheurs LCD (Liquid Crystal Display) sur le marché, non seulement dans différentes tailles (de 1 à 4 lignes de 6 à 80 caractères), mais aussi dans les caractéristiques techniques et les tensions de fonctionnement.

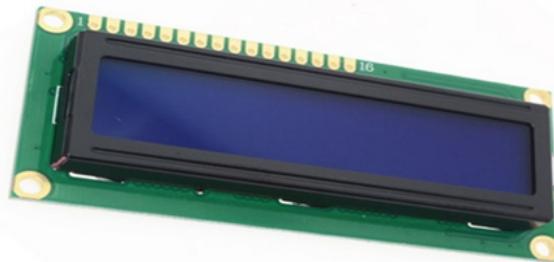


FIGURE 3.7 – Afficheur LCD.

### 3.2.1.6 Module I2C

Un I2C a été conçu par Philips au début des années 80 pour fournir un moyen simple de transférer des informations numériques entre les capteurs et des microcontrôleurs. Ce bus informatique (I2C) a l'avantage de n'avoir besoin que de deux connexions de signalisation :

**SCL (Serial Clock)** : horloge de communication.

**SDA (Serial Data)** : Signal de données (données série).

[27]

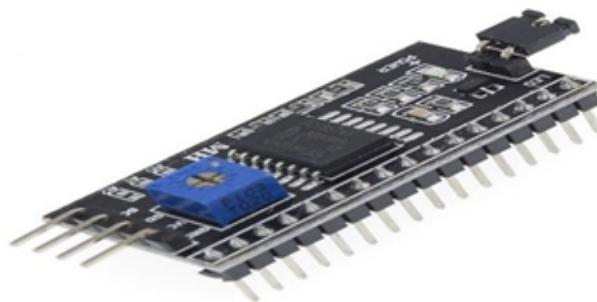


FIGURE 3.8 – Module I2C.

Dans notre cas, ce bus sera raccordé à l'écran LCD pour réduire le nombre des câbles à utiliser.

### 3.2.1.7 Plaque d'essai

Une plaque d'essai, souvent appelée breadboard ou protoboard, est utilisée pour construire des circuits temporaires, permettant de retirer et de remplacer facilement les composants. Il se compose d'un bloc en plastique avec cinq douilles connectées en rangées espacées de 2,54 mm. Il est constitué de deux matériaux : un isolant et un conducteur qui fonctionnent ensemble pour relier électriquement les trous dans une disposition horizontale ou verticale [28].

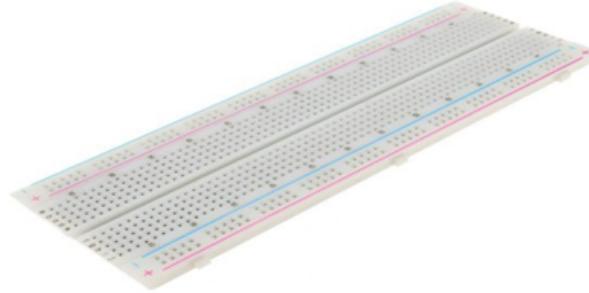


FIGURE 3.9 – Plaque d'essai.

### 3.2.1.8 Câbles

Les câbles électriques sont largement utilisés pour connecter différents appareils entre eux. Leur fonction principale est de permettre la transmission d'électricité d'un point à un autre de manière fiable et sécurisée.

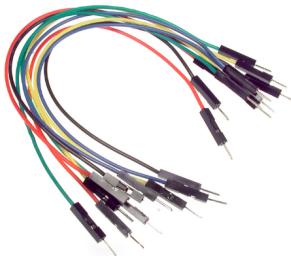


FIGURE 3.10 – Câble mâle-mâle.



FIGURE 3.11 – Câble mâle-femelle.

### Câble USB

Un câble USB est utilisé pour connecter une carte Arduino à un ordinateur, lui permettant de communiquer avec l'ordinateur et de l'alimenter à partir de ce dernier afin d'assurer 5 V.



FIGURE 3.12 – Câble USB.



FIGURE 3.13 – Câble Micro-USB.

## 3.2.2 Logiciels utilisés

### 3.2.2.1 Arduino IDE

Arduino IDE est un logiciel open source utilisé pour écrire et compiler le code dans la carte Arduino. Il est disponible pour les systèmes d'exploitation comme MAC, Windows, Linux et s'exécute sur la plateforme Java. Il contient deux parties de base : l'éditeur et le compilateur où le premier est utilisé pour écrire le code requis et l'autre

pour compiler et télécharger le code.

La figure suivante représente le logiciel Arduino IDE [29] :



FIGURE 3.14 – Arduino IDE.

### 3.2.2.2 Proteus

Proteus est un logiciel exclusif utilisé pour automatiser la conception électronique, exécuter des simulations et planifier la disposition des cartes de circuits imprimés. Il est utilisé par les ingénieurs et les techniciens pour développer des schémas et des impressions pour le prototypage rapide et la fabrication de PCB (Printed Circuit Board).

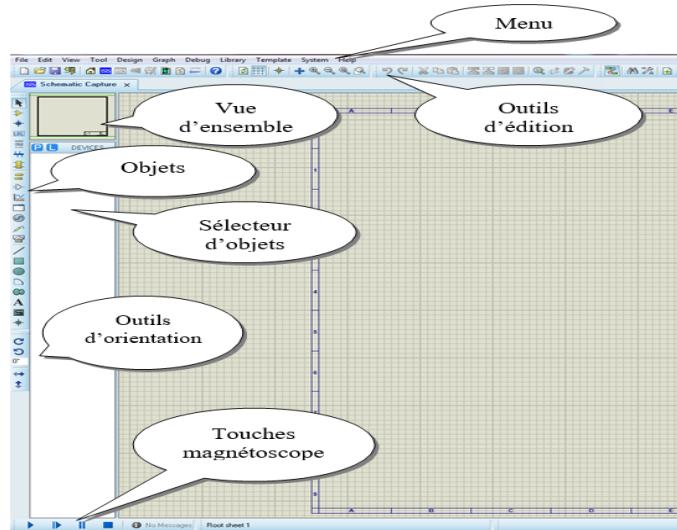


FIGURE 3.15 – Interface du logiciel Proteus.

### 3.2.2.3 ThingSpeak

ThingSpeak est une plateforme qui sert à analyser, visualiser et stocker les données en direct dans un serveur cloud, elle permet aussi d'envoyer des alertes aux utilisateurs par différents outils en utilisant des analyses MATLAB qui sont intégrées dans Thing-Speak.

Cette plateforme aide à créer les systèmes d'IOT sans avoir besoin d'installer des serveurs ou de constituer des applications Web [30].



FIGURE 3.16 – Logo de la plateforme ThingSpeak.

#### Caractéristiques de ThingSpeak :

Les atouts majeurs du logiciel incluent la communication inter-appareils, la gestion à distance, le suivi et la maintenance des objets connectés, et la possibilité d'accéder à tout moment aux données grâce à des fonctions d'import et d'export. La plateforme est accessible 24h/24 et 7j/7, offrant une grande souplesse d'utilisation.

Dans notre maquette on va connecter la carte Arduino à cette plateforme ThingSpeak, par la carte Wi-Fi ESP-32S, afin de visualiser les données qui viennent des capteurs : DHT11 et Anémomètre.

## 3.2.3 Simulation du prototype

### 3.2.3.1 Simulation du capteur DHT11 avec l'afficheur LCD

Le capteur DHT11 est connecté par la broche DATA à la pin 8 de la carte Arduino, et les deux broches SDA et SCL de l'afficheur LCD-I2C aux entrées analogiques A4 et A5 d'Arduino.

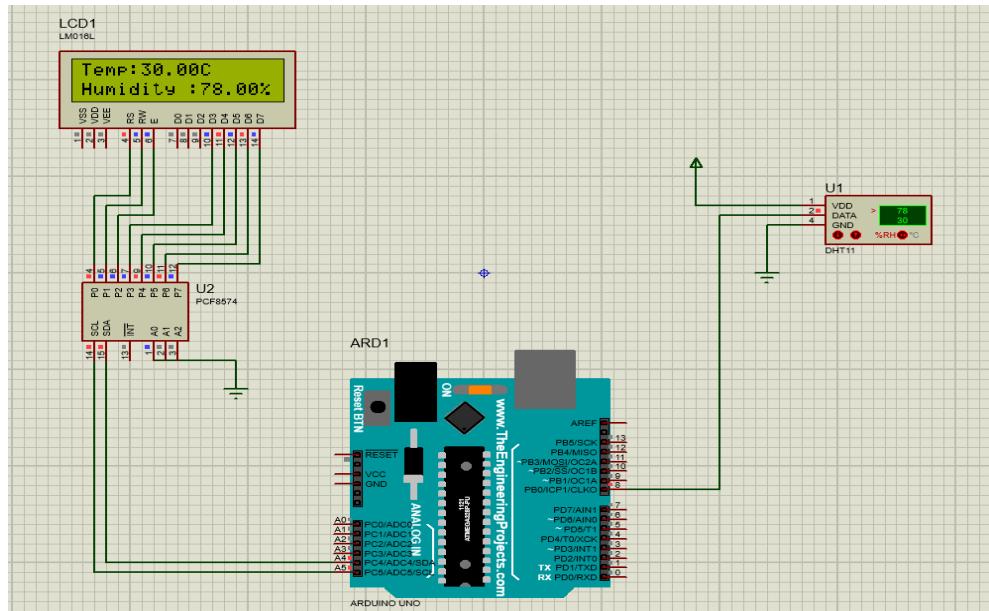


FIGURE 3.17 – Simulation du capteur DHT11 avec l'afficheur LCD-I2C.

### 3.2.3.2 Simulation du capteur de vitesse de vent (Moteur DC)

L'anémomètre qui est représenté sous forme d'un moteur DC dans le simulateur, où la broche (0V) du moteur est liée à la masse d'Arduino et la broche (5V) est liée à la pin A2 :

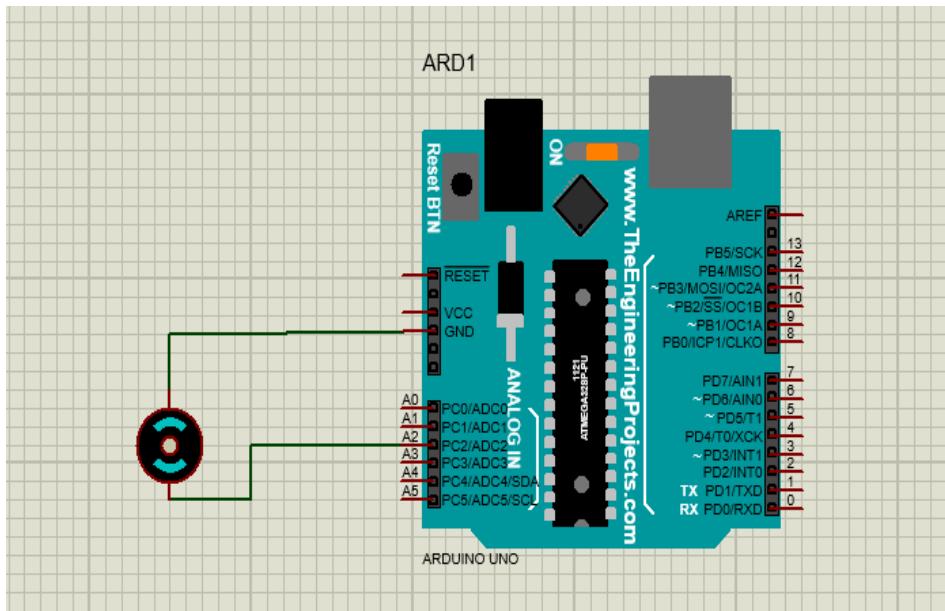


FIGURE 3.18 – Simulation de l'anémomètre.

### **3.2.3.3 Simulation de DHT11 et Anémomètre avec l'afficheur I2C-LCD(16x2)**

La figure ci-dessous représente la simulation des deux capteurs DHT11 et l'anémomètre, où la valeur de chaque capteur s'affiche sur l'afficheur I2C-LCD :

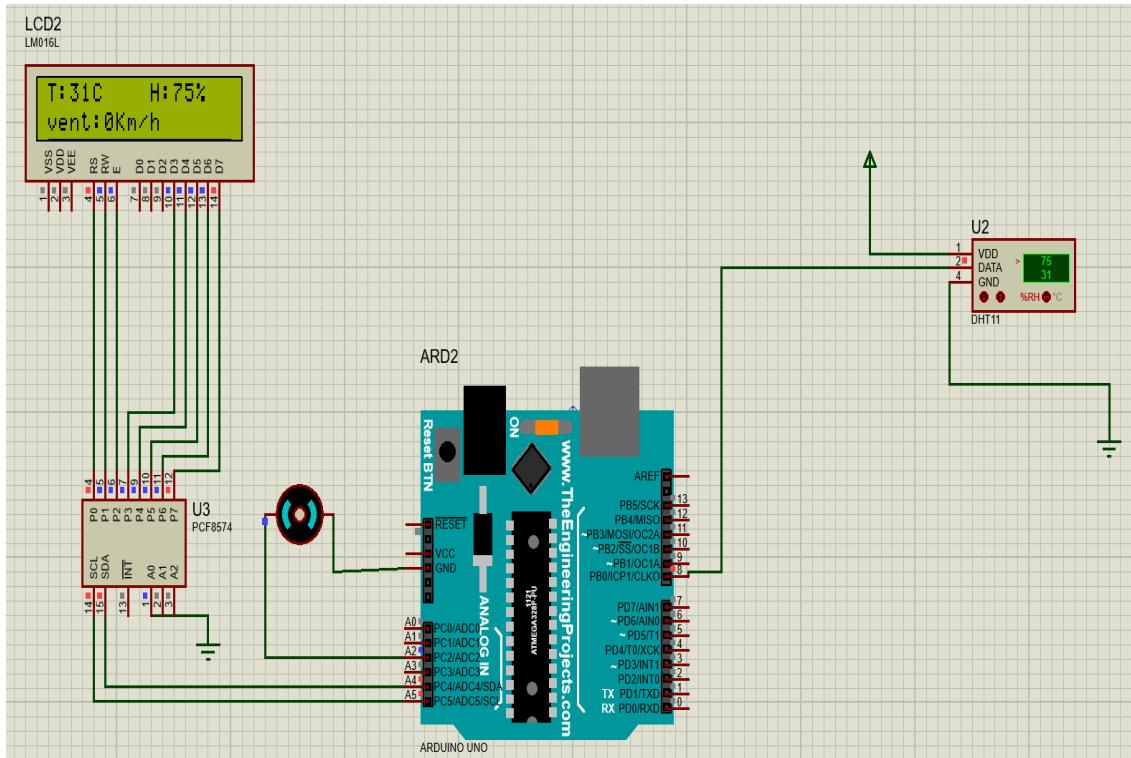


FIGURE 3.19 – Simulation de DHT11 et anémomètre avec l'afficheur LCD-I2C.

### 3.2.3.4 Simulation de prototype avec carte Wi-Fi ESP-32S

Pour faire communiquer l'ESP-32S à la carte Arduino, il faut brancher la broche Rx (Receive) de l'ESP-32S à la pin Tx (Transmit) d'Arduino et la broche Tx de l'ESP-32S à la pin Rx d'Arduino.

Cette carte Wi-Fi nous permet de réaliser un projet IoT qui va lier les données des capteurs à la plateforme ThingSpeak au but de les collecter et visualiser à distance :

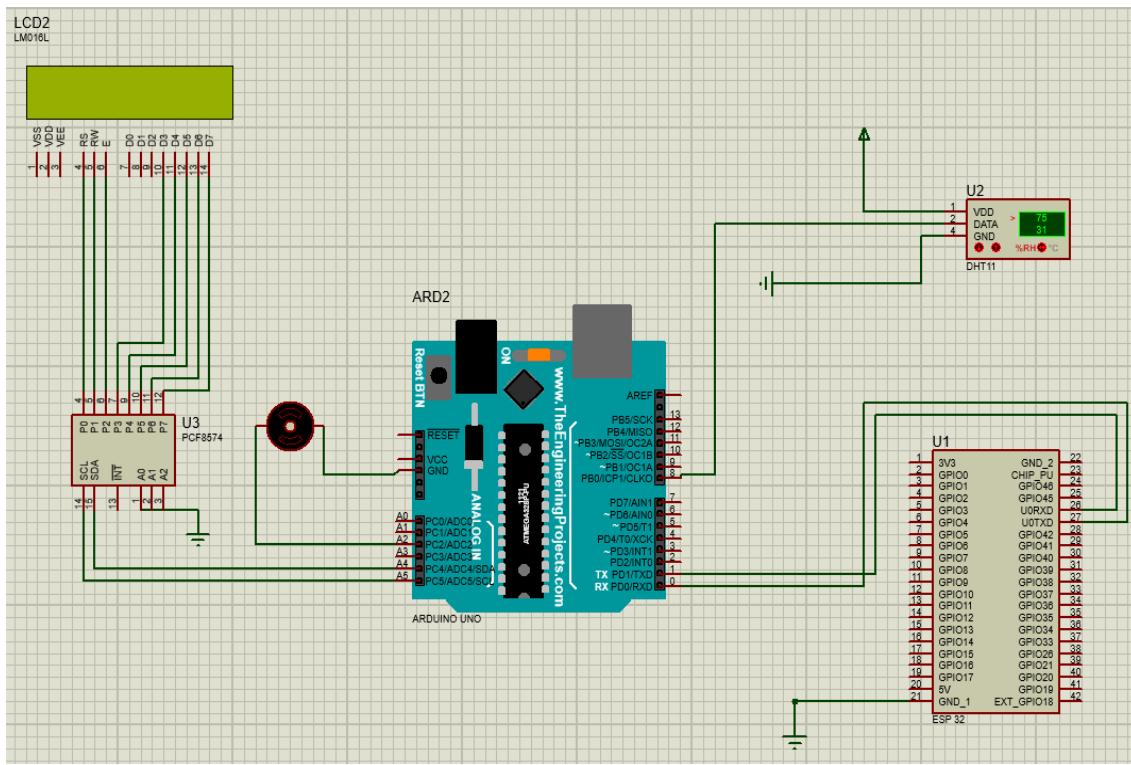


FIGURE 3.20 – Simulation de prototype avec la carte Wi-Fi ESP-32S.

## 3.3 Partie Machine Learning du prototype

### 3.3.1 Outils

#### 3.3.1.1 Python et Jupyter Notebook

Python est un langage de programmation préféré par les développeurs en raison de ses vastes fonctionnalités, de son applicabilité et de sa simplicité. Le langage de programmation Python convient le plus au ML en raison de sa large gamme de bibliothèques [31].

Jupyter Notebook, son nom vient des langages de programmation Julia, Python et R. C'est une application logicielle apparue en 2015, très utilisée par les data scientists grâce à sa facilité et simplicité, et le meilleur avantage c'est que les résultats d'exécution sont dans le même notebook, ce qui rend la programmation plus pratique.

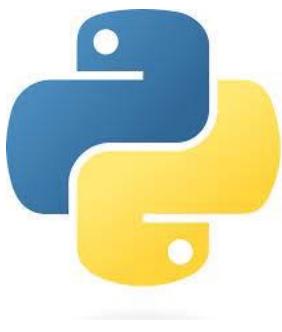


FIGURE 3.21 – Logo de Python.



FIGURE 3.22 – Logo de Jupyter.

### 3.3.1.2 Bibliothèques

#### Numpy :

Numpy est une librairie très utile dans programmation en python car il permet de manipuler des matrices, des tableaux (numpy.ndarray) de plusieurs dimensions, en général faire le calcul scientifique avec le langage de programmation python. Elle est très importante parce que en ML on a besoin des fois de manipuler notre dataset en utilisant les matrices.

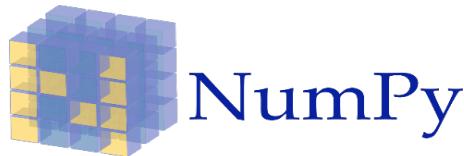


FIGURE 3.23 – Logo du package Numpy.

Pour avoir l'accès à la bibliothèque Numpy, il suffit de l'importer sur l'application Jupyter :

```
In [1]: import numpy as np
```

FIGURE 3.24 – Importation de la bibliothèque Numpy sur Jupyter.

#### Pandas :

Pandas est une librairie très populaire entre les data scientists, il permet de manipuler des Series qui sont tableaux unidimensionnels ou des DataFrame qui sont des structures de données bidimensionnelles. Il offre des fonctionnalités telles que le filtrage des données, le tri, la fusion, etc.



FIGURE 3.25 – Logo du package Pandas.

Pour avoir l'accès à la bibliothèque pandas, il suffit de l'importer sur l'application Jupyter :

```
In [2]: import pandas as pd
```

FIGURE 3.26 – Importation de la bibliothèque Pandas sur Jupyter.

### Matplotlib :

Matplotlib est une bibliothèque qui permet la visualisation des données sous forme des graphes, des courbes, etc.



FIGURE 3.27 – Logo du package matplotlib.

Pour avoir l'accès à la bibliothèque matplotlib et personnaliser des graphiques, il suffit de l'importer sur l'application Jupyter :

```
In [1]: import matplotlib.pyplot as plt
```

FIGURE 3.28 – Importation de la bibliothèque matplotlib sur Jupyter.

### SKlearn :

SKlearn est une bibliothèque spéciale au ML en python, elle offre une large gamme des algorithmes (des modèles) pour ML soit supervisé ou non supervisé. Elle offre aussi des métriques et des méthodes d'évaluation pour mesurer la performance des modèles.



FIGURE 3.29 – Logo du package SKlearn.

Pour avoir l'accès à la bibliothèque SKlearn, il suffit de l'importer sur l'application Jupyter :

```
In [4]: import sklearn
```

FIGURE 3.30 – Importation de la bibliothèque Sklearn sur Jupyter.

### 3.3.2 Méthodologie et préparation des données

#### 3.3.2.1 Création du dataset

Lorsqu'on parle du ML, le jeu de données ou dataset joue un rôle essentiel en fournissant les données nécessaires pour entraîner, évaluer et améliorer les modèles d'AI. Alors pour ce faire nous avons créé notre propre dataset à l'aide de deux capteurs, le premier a été employé afin de nous servir des mesures sur la température et l'humidité, le deuxième a été utilisé pour nous fournir des mesures sur la vitesse de vent.

Pour créer cette dataset, on a lié les capteurs à une carte Arduino, ainsi on a programmé en python un code sous forme d'une boucle qui collecte les données en temps réel provenant d'un appareil Arduino connecté via le port série COM 9 (Communication). Cette connexion est grâce à l'utilisation d'une bibliothèque nommée "serial" qui nous a permis d'élaborer une communication entre l'Arduino et le code en python. Ensuite les données sont traitées et stockées dans un DataFrame de la bibliothèque pandas en Python.

```
In [12]: import pandas as pd
import serial
import time

In [13]: ser = serial.Serial('COM9', 9600, timeout=4)

In [14]: data = pd.DataFrame(columns=['Temperature', 'RH', 'Ws', 'class'])
max_values = 500

while len(data) < max_values:
    donnees_arduino = ser.readline().decode('ascii')
    print(donnees_arduino)
    donnees_split = donnees_arduino.split(',')

    temperature = float(donnees_split[0])
    humidity = float(donnees_split[1].split(';')[0])
    wind_speed = float(donnees_split[1].split(';')[1])

    print("Temperature:", temperature)
    print("Humidity:", humidity)
    print("Wind Speed:", wind_speed)

    class_value = 1 if temperature > 30 or humidity < 50 or wind_speed > 15 else 0

    current_data = pd.DataFrame([[temperature, humidity, wind_speed, class_value]],
                                columns=['Temperature', 'RH', 'Ws', 'class'])
    data = data.append(current_data, ignore_index=True)

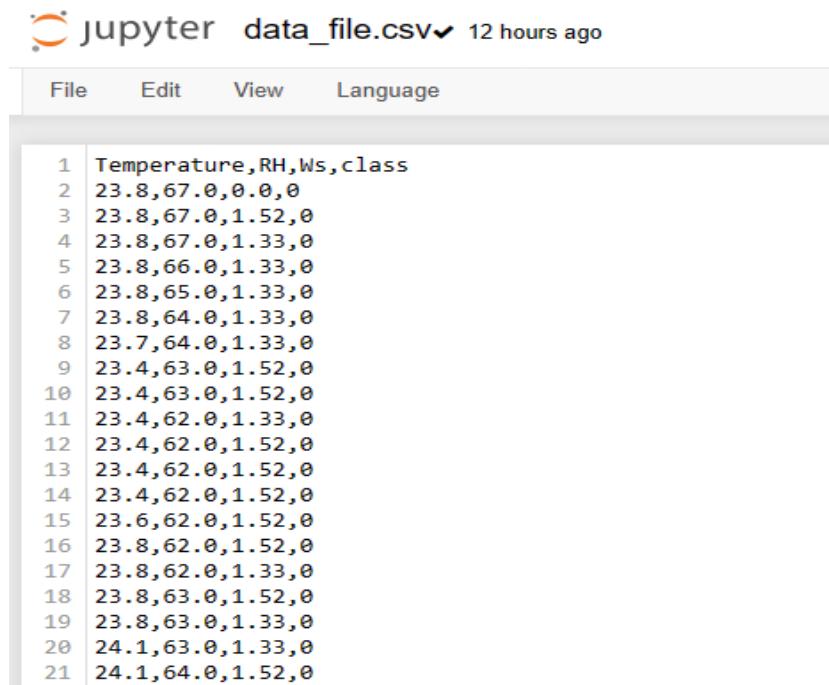
    data.to_csv('data_file.csv', index=False)

    time.sleep(4)
```

FIGURE 3.31 – Lignes du code pour créer le dataset.

Notre dataset a comme objectif de classifier les données collectées en deux catégories : incendie ou absence d'incendie. Alors, dans ce contexte, pour créer et organiser notre dataset, on s'est basé sur le travail réalisé par [6], et on s'est concentrés sur les données météorologiques (Température, Humidité, et vitesse de vent) ce qui nous a permis en fin de compte de choisir les conditions probables pour avoir un incendie.

On a ajouté à notre DataFrame une autre colonne appelée "class" qui va être réservée pour répondre aux conditions météorologiques soit par 1 lorsqu'on aura un incendie, ou par 0 lors d'absence d'incendie. Enfin toutes ces données seront stockées dans un fichier sous format CSV (Comma-Separated Values).

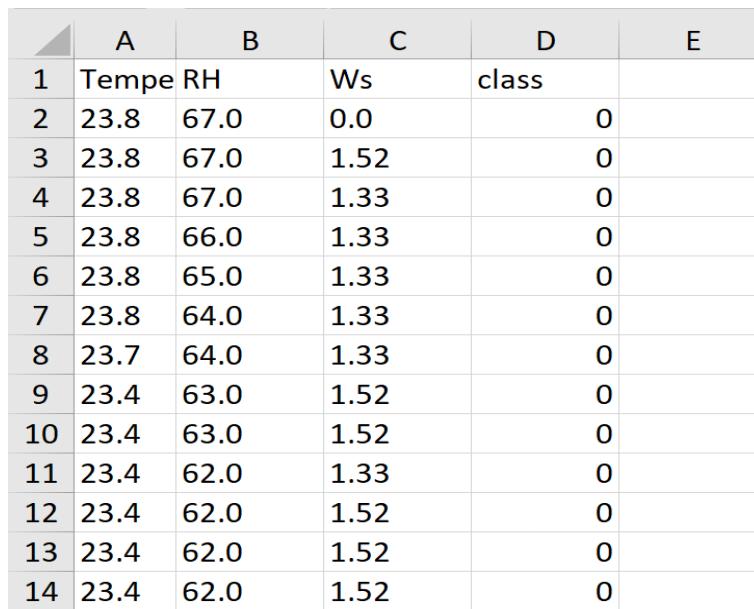


The screenshot shows a Jupyter Notebook interface with a tab titled "jupyter data\_file.csv" from 12 hours ago. The notebook has a menu bar with File, Edit, View, and Language. The main content area displays a CSV file with 21 rows of data. The columns are labeled 1 through 21, and the data includes Temperature, RH, Ws, and class values.

	1	Temperature, RH, Ws, class
2	23.8, 67.0, 0.0, 0	
3	23.8, 67.0, 1.52, 0	
4	23.8, 67.0, 1.33, 0	
5	23.8, 66.0, 1.33, 0	
6	23.8, 65.0, 1.33, 0	
7	23.8, 64.0, 1.33, 0	
8	23.7, 64.0, 1.33, 0	
9	23.4, 63.0, 1.52, 0	
10	23.4, 63.0, 1.52, 0	
11	23.4, 62.0, 1.33, 0	
12	23.4, 62.0, 1.52, 0	
13	23.4, 62.0, 1.52, 0	
14	23.4, 62.0, 1.52, 0	
15	23.6, 62.0, 1.52, 0	
16	23.8, 62.0, 1.52, 0	
17	23.8, 62.0, 1.33, 0	
18	23.8, 63.0, 1.52, 0	
19	23.8, 63.0, 1.33, 0	
20	24.1, 63.0, 1.33, 0	
21	24.1, 64.0, 1.52, 0	

FIGURE 3.32 – Base de données sous format CSV.

Pour avoir une organisation structurée des données, nous avons choisi de convertir notre dataset du format CSV au format Excel. Cela permet une visualisation et une manipulation facile des données tabulaires.



The screenshot shows an Excel spreadsheet with 14 rows of data. The columns are labeled A through E. Column A contains row numbers from 1 to 14. Columns B, C, D, and E contain the corresponding Temperature, RH, Ws, and class values respectively. The class column only contains the value 0.

	A	B	C	D	E
1	Tempe	RH	Ws	class	
2	23.8	67.0	0.0	0	
3	23.8	67.0	1.52	0	
4	23.8	67.0	1.33	0	
5	23.8	66.0	1.33	0	
6	23.8	65.0	1.33	0	
7	23.8	64.0	1.33	0	
8	23.7	64.0	1.33	0	
9	23.4	63.0	1.52	0	
10	23.4	63.0	1.52	0	
11	23.4	62.0	1.33	0	
12	23.4	62.0	1.52	0	
13	23.4	62.0	1.52	0	
14	23.4	62.0	1.52	0	

FIGURE 3.33 – Extrait de la base de données sous format Excel.

La base de données est dans le fichier Excel sous le nom "dataset", qui est composée de 3 "Features" qui sont (Temperature, RH, Ws), ainsi que la Target qui est "class".

```
In [223]: data=pd.read_excel("C:\\\\Users\\\\DELL\\\\Desktop\\\\dataset.xls")
data.head()
```

```
Out[223]:
```

	Temperature	RH	Ws	class
0	23.8	67.0	0.00	0
1	23.8	67.0	1.52	0
2	23.8	67.0	1.33	0
3	23.8	66.0	1.33	0
4	23.8	65.0	1.33	0

```
In [224]: y=data['class']
x=data.drop('class', axis=1)
```

FIGURE 3.34 – Lignes du code pour l'importation et le partitionnement de la base de données sur Jupyter.

### 3.3.2.2 Nettoyage des données

Tout d'abord, on a utilisé la fonction "isnull().sum()" pour calculer la somme des valeurs manquantes dans chaque colonne. Puis à l'aide de la fonction "dropna()" qui sert à supprimer les lignes qui contiennent au moins une valeur manquante, ainsi que "reset\_index(drop=True)" permet la réinitialisation de l'indice des lignes après la suppression.

La vérification avec la fonction "info()", nous a fourni une vue d'ensemble des données.

```
In [29]: data.isnull().sum()
Out[29]: Temperature      0
          RH              0
          Ws              0
          class           0
          dtype: int64

In [30]: data.shape
Out[30]: (495, 4)

In [31]: # Remove null or na values rows
          data = data.dropna().reset_index(drop=True)
          data.shape
Out[31]: (495, 4)

In [32]: # Final datatypes check
          data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 495 entries, 0 to 494
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype  
--- 
 0   Temperature  495 non-null    float64
 1   RH           495 non-null    float64
 2   Ws           495 non-null    float64
 3   class         495 non-null    int64  
dtypes: float64(3), int64(1)
memory usage: 15.6 KB
```

FIGURE 3.35 – Lignes du code pour nettoyer le dataset.

### 3.3.2.3 Visualisation des données

Dans cette partie, nous avons visualisé les trois "Features" et la "Target" avec le style "seaborn" qui sert à définir les couleurs et les polices des graphiques. La méthode "hist()" calcule et trace l'histogramme pour chaque colonne.

```
In [33]:
plt.style.use('seaborn')
data.hist(bins=50, figsize=(20,15), ec = 'b')
plt.show()
```

FIGURE 3.36 – Lignes du code pour visualiser les données

La figure ci-dessous montre la variation de la température de [23, 45] °C et l'humidité de [22,83] %, ce qui implique que le modèle d'apprentissage va être susceptible à plusieurs données et il va générer de bons scores lors de son apprentissage. Pourtant la vitesse de vent est fixée dans des petites valeurs pour ne pas impacter le modèle car elle n'a pas une grande importance au moment d'une inflammation, elle provoque que sa propagation.

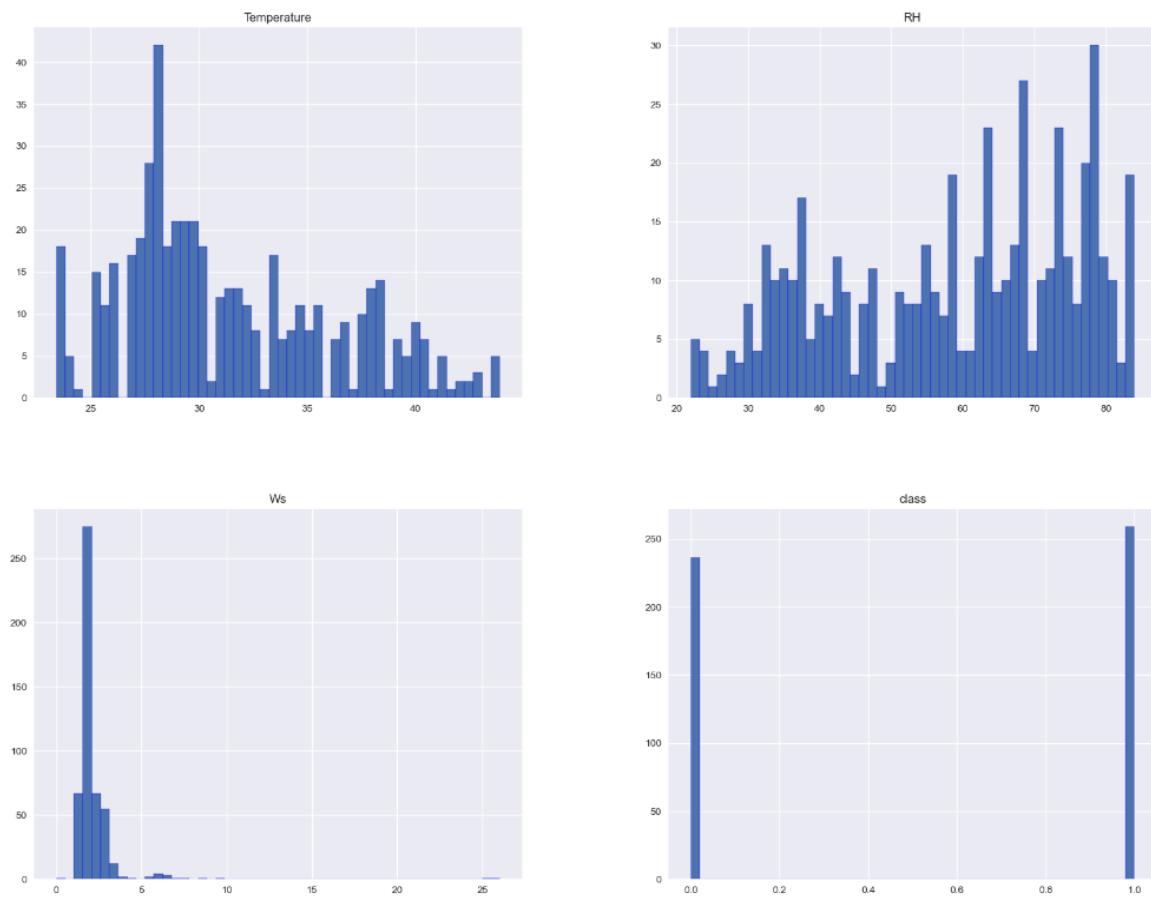


FIGURE 3.37 – Résultat de la visualisation.

Pour représenter la colonne "class" sous forme de graphique circulaire (diagramme en secteurs), ce code est fait afin de montrer le pourcentage de dominance de chaque classe.

La classe "1" est la plus dominante dans notre cas.

```
In [36]: percentage = data['class'].value_counts(normalize=True) * 100  
percentage  
Out[36]: 1    52.323232  
0    47.676768  
Name: class, dtype: float64
```

```
In [39]: | classeslabels = ["FIRE", "NOT FIRE"]  
plt.figure(figsize =(12, 7))  
plt.pie(percentage,labels = classeslabels,autopct='%.1f%%')  
plt.title ("Graphique circulaire des classes", fontsize = 15)  
plt.show()
```

Graphique circulaire des classes

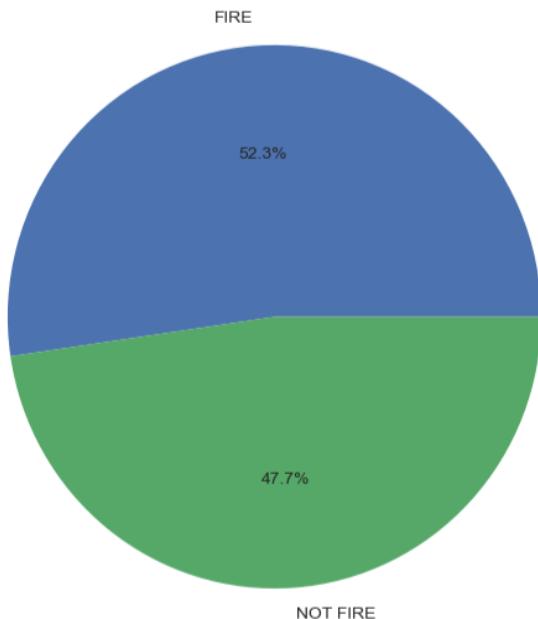


FIGURE 3.38 – Code de dominance et son résultat graphique.

#### 3.3.2.4 Partitionnement des données

A l'aide de la fonction "`train_test_split`" qui est importé de la bibliothèque "`sklearn.model_selection`", on a divisé notre jeu de données en deux catégories : Les 80 % des données sont utilisées pour l'entraînement et les 20% restants sont employés pour les tests , la valeur "`random_state=2`" sert à garantir la reproductibilité des résultats.

```
In [ ]: from sklearn.model_selection import train_test_split  
  
In [86]: x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=2)
```

FIGURE 3.39 – Importation des bibliothèques nécessaires pour le partitionnement des données.

### 3.3.3 Modèles de classification

Nous avons choisi les modèles de classification pour objectif de prévoir la classe ou la catégorie à laquelle appartient les nouvelles entrées des capteurs, parmi ces modèles on a :

#### 3.3.3.1 Arbres de décision

Les arbres de décision sont un algorithme de ML supervisé, il est facile à comprendre et à interpréter. Le but est de construire un modèle qui prévoit la valeur d'une variable cible "Target" en apprenant des règles de décision simples à l'aide des caractéristiques "Features" sous la forme d'une arborescence [32]. Chaque nœud représente un test, chaque branche représente une conséquence, et chaque feuille représente une classe.

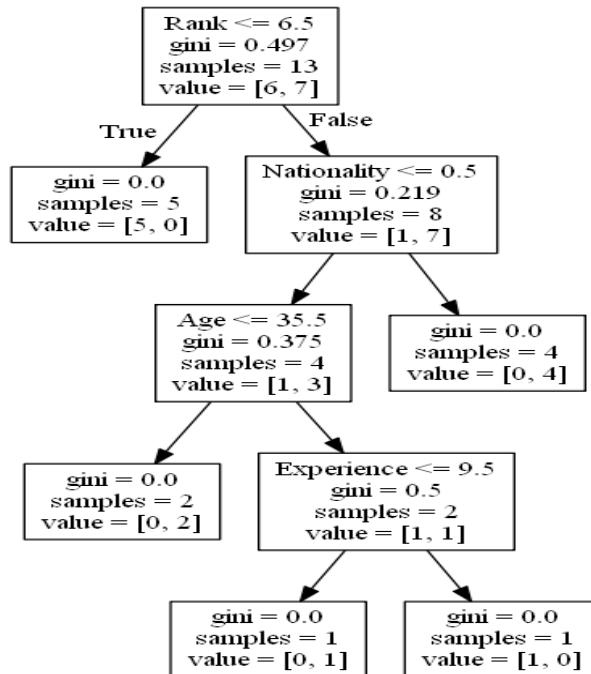


FIGURE 3.40 – Exemple des arbres de décision en ML.

Pour diviser les données en sous-ensembles plus homogènes, les caractéristiques les plus significatives doivent être sélectionnées. Pour cette raison, nous devons mesurer deux mesures de pureté : Gini ou Entropie (il existe d'autres critères, tels que la profondeur maximale et le nombre minimal d'arbres).

L'indice Gini mesure l'impureté de l'ensemble de données, il est compris entre 0 et 1, tel que le 1 indique que l'ensemble est dans son impureté maximale. Pour calculer cet indice il faut appliquer les lois suivantes :

$$Gini = \left( \frac{Sg}{S} \right) \times (E(Sg)) + \left( \frac{Sd}{S} \right) \times (E(Sd)) \quad (3.2)$$

Avec Sg et Sd sont le nombre des sous-ensembles gauches et droits, et E(Sg) et E(Sd) sont les fonctions coût gauches et droites.

La fonction coût E(S) :

$$E(S) = 1 - \sum P^2(ci) \quad (3.3)$$

Tel que "ci" est le nombre des échantillons, et S le nombre de tous les échantillons.

$$Avec : P(ci) = \frac{ci}{S} \quad (3.4)$$

L'indice entropie mesure la pureté d'ensemble de données, il est compris entre 0 et 1, tel que le 0 indique que l'entropie est minimale, et le gain d'information est comme meilleure, et s'il y a une division des données égale, l'entropie vaille 1. Pour calculer cet indice il faut appliquer les lois suivantes :

$$IG = H(S) - \sum P(Sjk)H(Sjk) \quad (3.5)$$

Avec  $H(Sjk)$  est la fonction d'entropie des sous-ensembles et leur probabilités  $P(Sjk) = \frac{Sjk}{S}$ , tel que la relation suivante présente la loi d'entropie  $H(S)$  :

$$H(S) = - \sum P(ci) \times \log(ci)$$

Avec :

$$P(ci) = \frac{ci}{S} \quad (3.6)$$

Il faut importer la bibliothèque du modèle "DecisionTreeClassifier" qui a été saisie par "sklearn.tree", ainsi que "cross\_val\_score" qui est importé de la sélection "sklearn.model\_selection" pour la validation croisée (consiste à diviser l'ensemble de données en plusieurs parties "folds") pour évaluer les performances du modèle.

```
In [5]: from sklearn.tree import DecisionTreeClassifier  
from sklearn.model_selection import cross_val_score
```

FIGURE 3.41 – Importation des bibliothèques nécessaires pour le modèle des arbres de décision.

On a choisi "max\_depth" et le critère "Gini" comme hyper-paramètres pour savoir les bons scores moyennant de la fonction "cross\_val\_score" avec "cv=5", qui sert à répartition de notre dataset sur 5 parties, pour une plage de [1,50] de "max\_depth". On remarque qu'à partir de 2 le score d'entraînement atteint 100%.

```
In [36]:  
val_score = []  
for max_depth in range(1, 50):  
    tree = DecisionTreeClassifier(criterion='gini', max_depth=max_depth)  
    score = cross_val_score(tree, x_train, y_train, cv=5, scoring='accuracy').mean()  
    val_score.append(score)  
  
plt.plot(val_score)
```

```
Out[36]: [matplotlib.lines.Line2D at 0x244a95316c0]
```

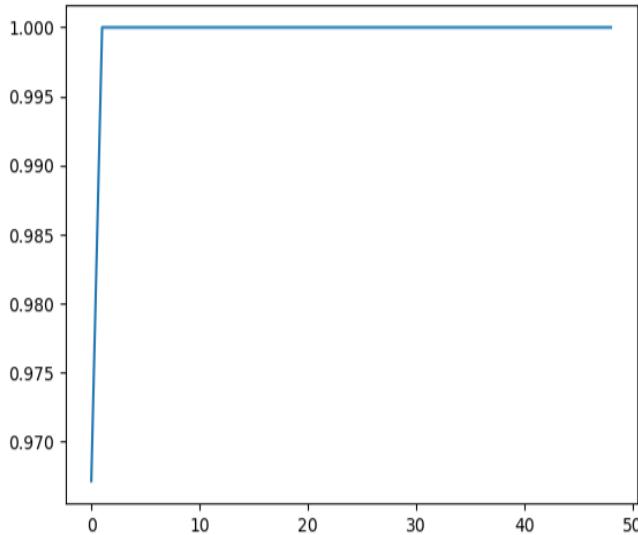


FIGURE 3.42 – Courbe des scores du modèle des arbres de décision.

On a choisi "max\_depth=5" pour notre modèle comme la courbe ci-dessus a montré, et la fonction "fit()" qui permet d'entrainer le modèle sur les données, et on a eu 100% pour l'entraînement et 97% du test.

```
In [8]: x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=2)
tree=DecisionTreeClassifier(criterion='gini',max_depth=5,random_state=2)
tree.fit(x_train, y_train)
tree.score(x_train, y_train)
```

```
Out[8]: 1.0
```

```
In [9]: tree.score(x_test, y_test)
```

```
Out[9]: 0.9797979797979798
```

FIGURE 3.43 – Lignes du code du modèle des arbres de décision.

### 3.3.3.2 Forêt aléatoire

Une forêt aléatoire est un algorithme du ML supervisé qui est construit à partir d'algorithmes d'arbre de décision. Cet algorithme établit le résultat en fonction des prévisions des arbres de décision. Il prévoit en prenant la moyenne de la sortie de divers arbres. L'augmentation du nombre d'arbres augmente la précision du résultat.

Parmi les avantages de cet algorithme qu'il est plus précis que l'algorithme de l'arbre de décision, et il résout le problème de over-fitting ou le surapprentissage dans les arbres de décision. (Over-fitting est un phénomène qui indique qu'un modèle de ML a ajusté ses données d'apprentissage et a perdu sa capacité à généraliser correctement à de nouvelles données) [33].

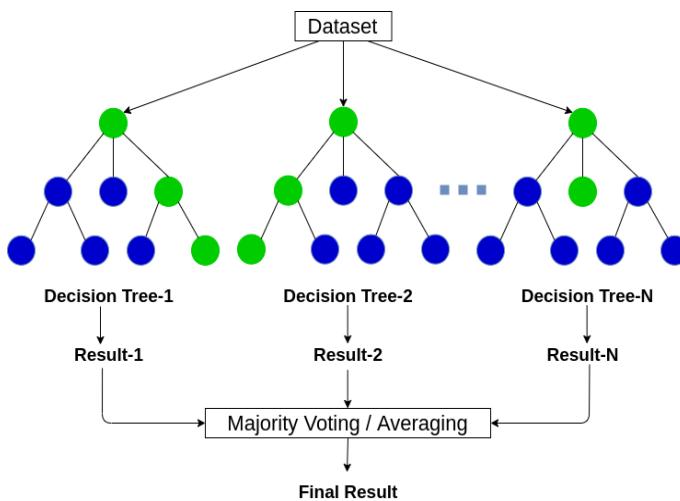


FIGURE 3.44 – Schéma explicatif du modèle de la forêt aléatoire.

Il faut importer la bibliothèque du modèle "RandomForestClassifier" qui a été saisie par "sklearn.ensemble", ainsi que "cross\_val\_score" sont importés de "sklearn.model\_selection".

```
In [1]: from sklearn.ensemble import RandomForestClassifier  
from sklearn.model_selection import cross_val_score
```

FIGURE 3.45 – Importation des bibliothèques nécessaires pour le modèle de la forêt aléatoire.

On a choisi "n\_estimators" et le critère "entropy" comme hyper-paramètres dans la validation croisée. On remarque que le score d'entraînement atteint 100% avec certaines valeurs.

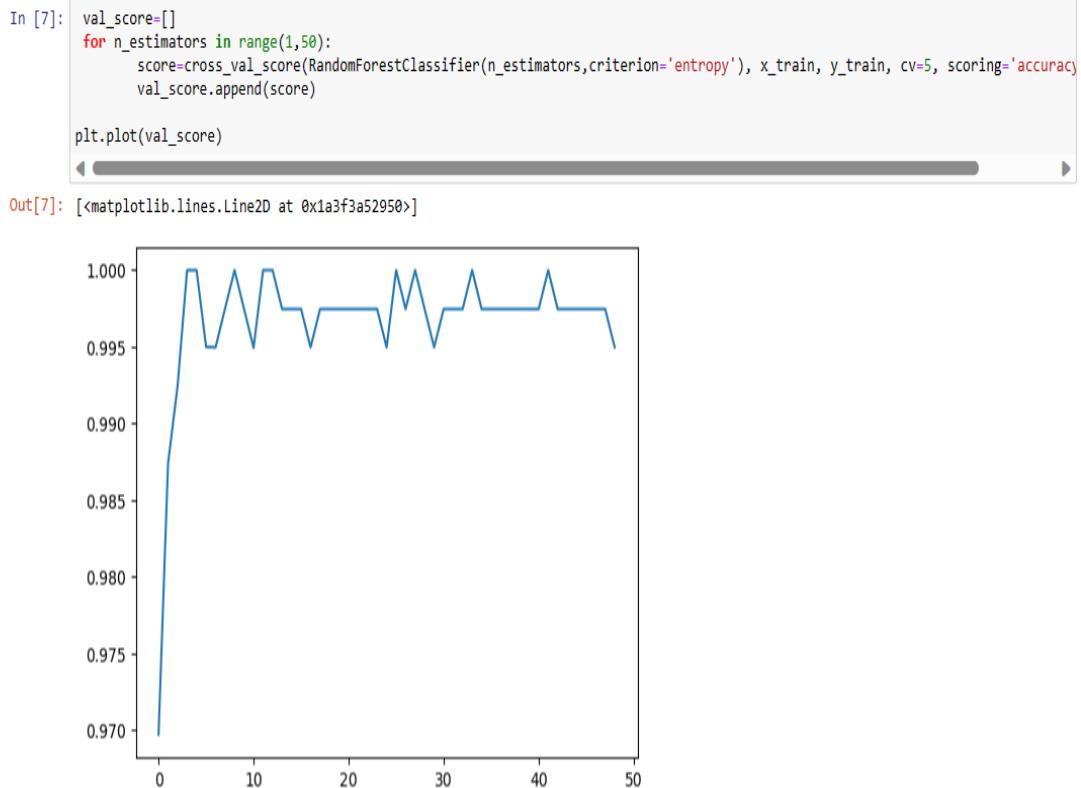


FIGURE 3.46 – Courbe des scores du modèle de la forêt aléatoire.

On a choisi le critère "entropy" pour notre modèle et le "n\_estimators=24" comme la courbe ci-dessus a montré et on a eu 100% pour l'entraînement et 97% du test.

```
In [225]: RFC=RandomForestClassifier(criterion='entropy', n_estimators=24, random_state=2)

In [207]: RFC.fit(x_train, y_train)
RFC.score(x_train, y_train)

Out[207]: 1.0

In [208]: RFC.score(x_test, y_test)

Out[208]: 0.9797979797979798
```

FIGURE 3.47 – Lignes du code du modèle de la forêt aléatoire.

### 3.3.3.3 Classifieur Bayésien Naïf

Le classifieur Bayésien Naïf est l'un des fameux algorithmes du ML supervisé utilisé pour résoudre les problèmes de classification. Cette technique est basée sur le théorème de Bayés avec une forte indépendance des hypothèses. Ce théorème emploie une connaissance préalable des termes associés pour estimer la probabilité d'un événement. La formule générale du théorème de Bayés est la suivante :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

**A** et **B** sont des événements

**P(A)** est la probabilité d'observer l'événement A

**P(B)** est la probabilité d'observer l'événement B

**P(A|B)** : représente la probabilité de l'événement A sachant l'événement B

**P(B|A)** : représente la probabilité de l'événement B sachant l'événement A

On a appliqué ce modèle "GaussianNB" à notre dataset, afin de l'entraîner et l'évaluer. on a importé le modèle de la bibliothèque "naive\_bayes".

```
In [2]: from sklearn.naive_bayes import GaussianNB
```

FIGURE 3.48 – Importation de la bibliothèque du modèle du Bayésien Naïf.

On a créé un objet de GaussianNB appelé "clf". Après on a utilisé la fonction "fit "pour entraîner le classificateur en adaptant les données d'entraînement "x\_train" avec les étiquettes d'entraînement "y\_train".

```
In [4]: clf = GaussianNB()
In [5]: clf.fit(x_train,y_train)
Out[5]: GaussianNB()
```

FIGURE 3.49 – Entrainement du modèle du Bayésien Naïf.

la figure ci-dessous présente les scores obtenus par cette méthode :

```
In [9]: clf.score(x_train, y_train)
Out[9]: 0.9545454545454546
In [10]: clf.score(x_test, y_test)
Out[10]: 0.9292929292929293
```

FIGURE 3.50 – Scores du modèle du Bayésien Naïf.

### 3.3.3.4 k plus proches voisins

L'algorithme KNN (K-Nearest Neighbors) qui signifie les k plus proches voisins, est une méthode qui nous donne la possibilité de résoudre les problèmes de classification. Il est classé comme étant un algorithme du ML supervisé simple et facile à mettre en œuvre, il nous permet de faire des prévisions sur des variables qualitatives et quantitatives.

Pour exercer ce modèle de ML sur notre dataset, on a tout d'abord importer les bibliothèques nécessaires pour notre démarche, le modèle KNN a été sélectionné de "sklearn.neighbors", ensuite "cross\_val\_score" est importé de "sklearn.model\_selection".

```
In [3]: from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
```

FIGURE 3.51 – Importation des bibliothèques du modèle KNN.

A ce point on a effectué une validation croisée pour évaluer les performances du modèle KNN avec différentes valeurs de k, puis on a tracé un graphique des scores de validation moyens en fonction de K la figure ci-dessous la présente.

```
In [9]: val_score=[]
for k in range(1,50):
    score=cross_val_score(KNeighborsClassifier(k), x_train, y_train, cv=5, scoring='accuracy').mean()
    val_score.append(score)

plt.plot(val_score)
```

Out[9]: [`<matplotlib.lines.Line2D at 0x2147773ce0>`]

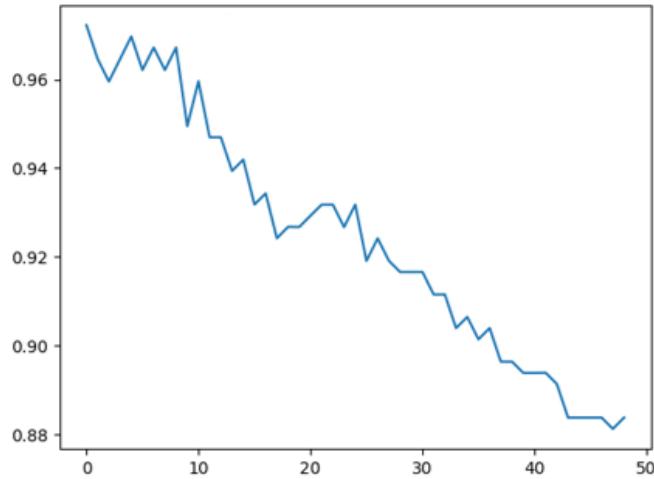


FIGURE 3.52 – Courbe des scores du modèle KNN.

A cette étape on a créé un objet du modèle KNN sous le nom "clfK", tel que dans la classe "KNeighborsClassifier" de scikit-learn, le paramètre qui spécifie le nombre de voisins les plus proches est appelé "n\_neighbors", nous avons utilisé la méthode "fit" afin d'ajuster le classifieur "KNeighborsClassifier" sur les données d'entraînement. Les scores obtenus par cette technique sont indiqués dans la figure ci-joint :

```
In [215]: ClfK=KNeighborsClassifier(n_neighbors=1)
```

```
In [216]: ClfK.fit(x_train, y_train)
ClfK.score(x_train, y_train)
```

Out[216]: 0.9873737373737373

```
In [217]: ClfK.score(x_test, y_test)
```

Out[217]: 0.9696969696969697

FIGURE 3.53 – Lignes du code du modèle KNN.

**Récapitulatif :**

Modèle	arbres de décision	forêt aléatoire	Bayésien Naïf	KNN
Score d'entraînement	100%	100%	95.45%	98.73%
Score du test	97.98%	97.98%	92.92%	96.97%

TABLE 3.3 – Comparaison des scores des différents modèles

Après une comparaison entre les 4 modèles, on déduit que les arbres de décision et la forêt aléatoire ont eu le meilleur score d'entraînement et du test aussi.

## 3.4 Design du prototype en 3D

Voici ci-joint le design de la forêt en 3D :



FIGURE 3.54 – Design du prototype en 3D.

## 3.5 Réalisation du prototype

### 3.5.1 Schéma fonctionnel du système proposé

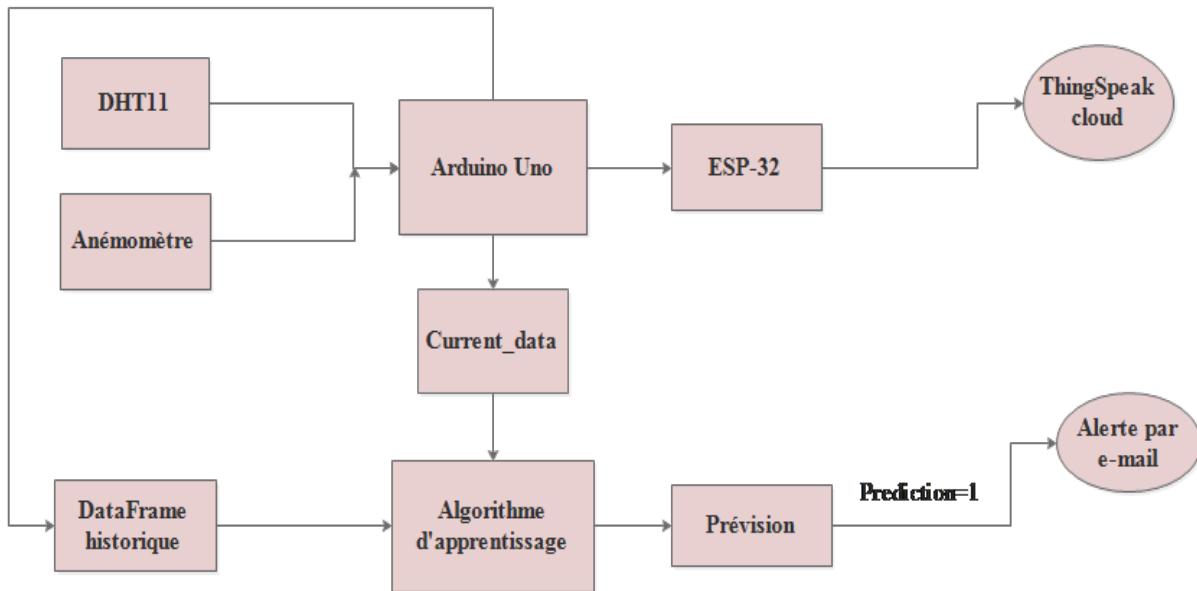


FIGURE 3.55 – Schéma fonctionnel du système proposé.

### 3.5.2 Réalisation réelle

Les figures ci-dessous montrent l'installation d'une station de surveillance dans la forêt, équipée de divers composants électroniques tels que le DHT11, l'anémomètre, l'Arduino Uno, l'ESP-32S et l'afficheur LCD. Ces composants sont interconnectés et permettent de surveiller en temps réel les fluctuations de trois paramètres essentiels : la température, l'humidité et la vitesse du vent. Les données collectées sont régulièrement enregistrées dans le programme du ML à intervalles d'une seconde, afin de réaliser des prévisions en temps réel. En cas de prévision d'incendie, le service des pompiers sera alerté par e-mail.



FIGURE 3.56 – Prototype de la forêt en réalité.



FIGURE 3.57 – Prototype de la forêt avec la station de surveillance en réalité.

L'afficheur LCD présente en temps réel les mesures de température en degrés Celsius, d'humidité en pourcentage et de vitesse du vent en kilomètres par heure, actualisées chaque seconde.

Les figures ci-dessous illustrent une augmentation des valeurs de température et d'humidité. Ces valeurs seront mises à jour à la fois sur l'afficheur LCD et sur la plateforme ThingSpeak (comme indiqué dans la figure 3.60).

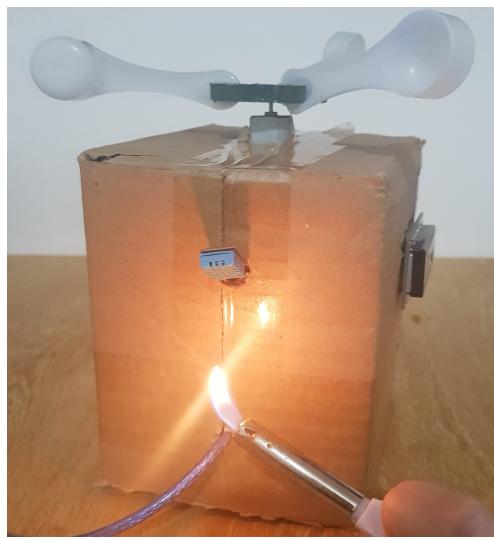


FIGURE 3.58 – Variation de la température et d’humidité.

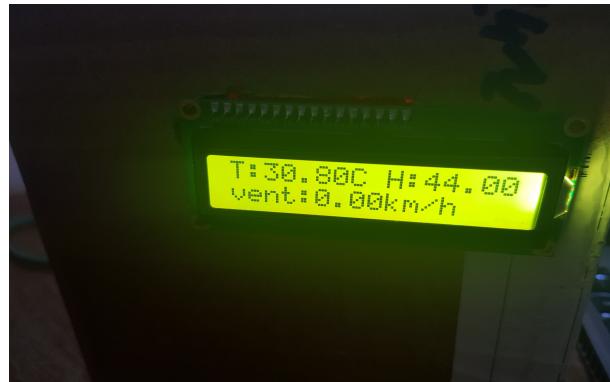


FIGURE 3.59 – Augmentation de température et diminution d’humidité.

### 3.5.3 Connexion à ThingSpeak

Nous avons créé un compte sur ThingSpeak, puis un canal où les données sont stockées, ensuite à l'aide de la carte Wi-Fi ESP-32S on a pu assurer la connexion d'Arduino au réseau Wi-Fi. Cette étape nous a permis de visualiser à distance les données sous forme des courbes en temps réel.

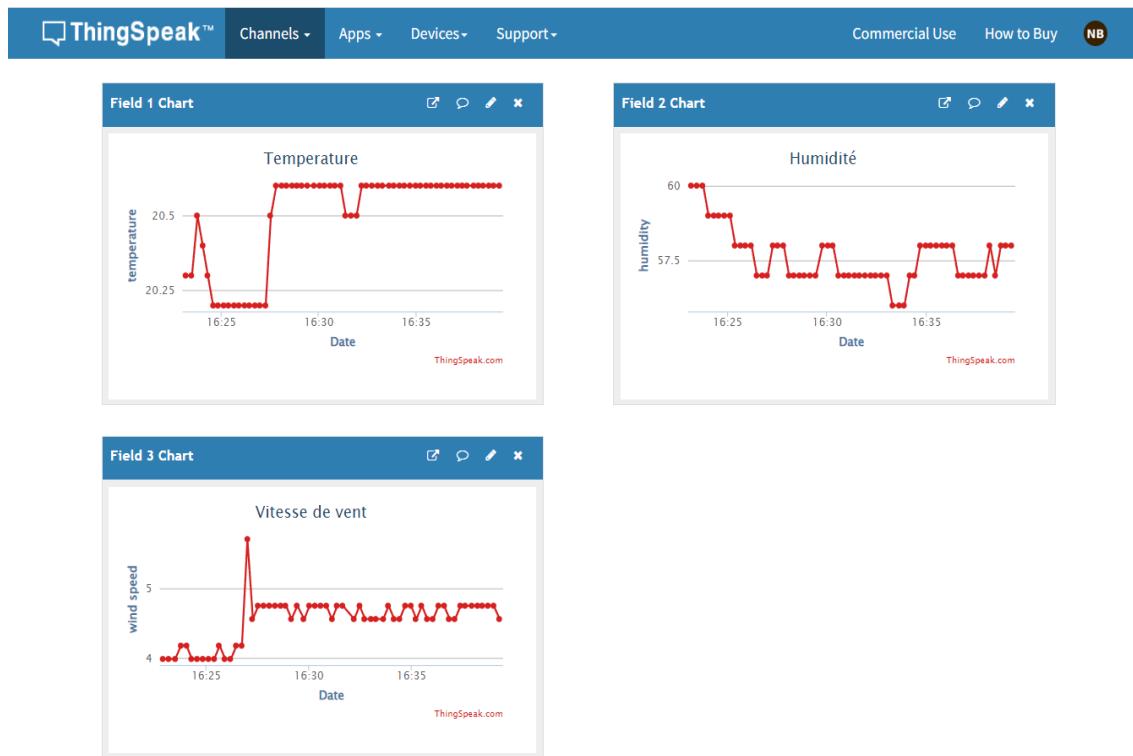


FIGURE 3.60 – Visualisation des données sur ThingSpeak.

### 3.5.4 Communication Arduino-Python et Alerte

Le dispositif Arduino est connecté via le port série (COM9) avec une vitesse de transmission 9600 bits/s.

Dans le code suivant nous avons "email\_sender", "email\_password" et "email\_receiver" pour entrer les informations d'authentification, puis dans une boucle "while True" qui est une boucle infinie, nous permet l'exécution du code en continue, et à l'aide de la bibliothèque pandas on a pu créer le DataFrame "current\_data" à partir des valeurs actuelles de température, d'humidité et de vitesse du vent. En outre dans la même boucle "RFC.predict(current\_data)" utilise le modèle "RandomForestClassifier" préalablement défini pour effectuer une prévision à l'aide des données extraites.

### 3.5. RÉALISATION DU PROTOTYPE

---

```
: ser = serial.Serial('COM9', 9600)

: # Configuration des informations d'authentification pour l'envoi d'e-mails
email_sender = 'firepredection@gmail.com'
email_password = 'qavm wroh ooai kzzp'
email_receiver = 'popompiers9@gmail.com'

while True:
    donnees_arduino = ser.readline().decode('ascii')
    print(donnees_arduino)
    donnees_split = donnees_arduino.split(',')

    temperature = donnees_split[0]
    humidity = donnees_split[1].split(';')[0]
    wind_speed = donnees_split[1].split(';')[1]

    print("Temperature : ", temperature)
    print("Humidity : ", humidity)
    print("Wind Speed : ", wind_speed)

    # Création d'un DataFrame avec les données du capteur actuel
    current_data = pd.DataFrame([[float(temperature), float(humidity), float(wind_speed)]], columns=['Temperature', 'RH', 'Ws'])

    # Comparaison des données actuelles avec les données existantes
    prediction = RFC.predict(current_data)

    # Affichage de la prédition
    print("Prédition : ", prediction)
```

FIGURE 3.61 – Lignes du code pour afficher les données du moniteur série avec leurs prévisions.

L'exécution de ce code nous affiche la température, l'humidité et la vitesse du vent en temps réel avec leur prévision instantanément :

```
29.30,58.00;4.37
Temperature : 29.30
Humidity : 58.00
Wind Speed : 4.37

Prédiction : [0]
30.20,50.00;4.37
Temperature : 30.20
Humidity : 50.00
Wind Speed : 4.37

Prédiction : [1]
31.30,45.00;4.18
Temperature : 31.30
Humidity : 45.00
Wind Speed : 4.18

Prédiction : [1]
```

FIGURE 3.62 – Données en temps réel avec leurs classes.

Et toujours dans la même boucle on assure une alerte si `prediction==1`, le code crée un objet "EmailMessage" pour composer un e-mail contient "fire", et les informations

d'expéditeur, de destinataire, de sujet et de contenu de l'e-mail sont définies. Ensuite, une connexion sécurisée SSL est établie avec le serveur SMTP de Gmail et l'e-mail est envoyé.

```
if prediction == 1:  
    subject = 'Alert'  
    body = 'fire'  
    em = EmailMessage()  
    em['From'] = email_sender  
    em['To'] = email_receiver  
    em['Subject'] = subject  
    em.set_content(body)  
    context = ssl.create_default_context()  
    with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=context) as smtp:  
        smtp.login(email_sender, email_password)  
        smtp.send_message(em)  
  
    time.sleep(1)
```

FIGURE 3.63 – Lignes du code de l'e-mail.

Voici l'e-mail envoyé à l'aide de ces instructions-là :

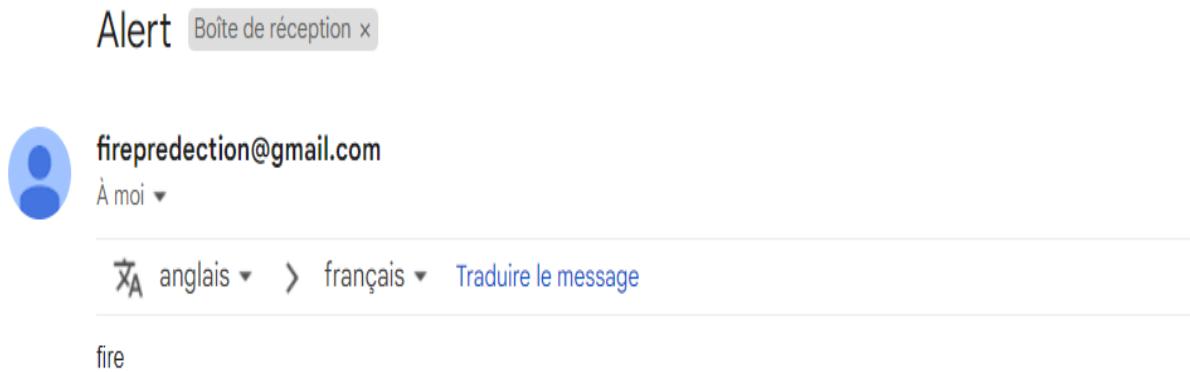


FIGURE 3.64 – E-mail envoyé.

## 3.6 Conclusion

Dans ce chapitre, nous avons discuté dans la partie électronique des différents composants tels que la carte Arduino Uno, la carte Wi-Fi ESP-32S, le DHT11, l'anémomètre, etc. Ensuite, nous avons défini les logiciels utilisés tels que l'Arduino IDE, Proteus et la plateforme ThingSpeak. De plus, nous avons simulé tout le circuit électronique afin

d'éviter les problèmes de connexion ultérieurs.

En ce qui concerne la partie ML, nous avons utilisé le langage Python pour sa facilité de programmation. À l'aide des capteurs, nous avons pu collecter les données et construire notre propre base de données, sur laquelle nous avons entraîné différents algorithmes de classification en ML supervisé. Enfin, nous avons choisi le meilleur modèle qui a obtenu le meilleur score.

En clôturant ce chapitre, nous avons réalisé un design en 3D et avons concrètement créé une forêt accompagnée d'une station de surveillance. Cette station comprend tous les microcontrôleurs et les capteurs intelligents qui ont permis de collecter les données et de les visualiser sur la plateforme ThingSpeak. De plus, ces données ont été intégrées dans un programme de prévision en temps réel grâce à la communication entre Arduino et Python. En cas de prévision d'incendie, le service des pompiers recevra une alerte par e-mail.

---

# CONCLUSION GÉNÉRALE ET PERSPECTIVES

L'objectif principal de notre projet de fin d'étude est de réaliser un système intelligent basé sur l'IOT et AI afin de prévoir les incendies des forêts.

Le projet qu'on a réalisé est capable de collecter, analyser et interpréter les données importées par les capteurs (DHT11 et anémomètre) qui sont placés dans une station près de la forêt.

Pour ce but, on a divisé notre projet en deux grandes parties (électronique et ML), la première est consacrée à l'IOT, en connectant les deux capteurs à la carte Arduino Uno et la carte Wi-Fi ESP-32S, ce qui nous permet de rassembler les données en temps réel et les visualiser sur la plateforme connue sous le nom « ThingSpeak », la deuxième partie concerne l'entraînement des différents modèles de classification supervisé tels que la forêt aléatoire, les arbres de décision, etc, sur nos données historiques (créer à l'aide des deux capteurs). Enfin on a choisi le meilleur modèle en se basant sur le score. Par la suite, à l'aide d'une communication entre l'Arduino et l'application de ML, on a pu envoyer une alerte par email au service pompiers.

Nos résultats montrent sans équivoque que l'utilisation du ML dans les applications électroniques permet d'améliorer considérablement les performances, la fiabilité et l'efficacité des systèmes électroniques.

Concernant l'état future de notre projet, pour rendre notre modèle de prévision plus puissant, on peut ajouter un capteur d'oxygène pour agrandir la taille de dataset, ce qui va influencer positivement sur l'apprentissage des modèles et donne les meilleures performances. De plus, nous pouvons envoyer une alerte indiquant l'endroit où se trouve l'incendie. Par ailleurs, nous pouvons installer notre système dans un véhicule aérien pour couvrir les prévisions sur une large zone.

Nous avons pu accroître notre compréhension en électronique grâce à ce travail. En

outre, ce projet nous a fourni une occasion pour développer nos aptitudes et acquérir de nouvelles expériences tels que la programmation avec Arduino IDE, ainsi qu'il nous a permis de se familiariser avec Python et le ML supervisé.

Avec tous ces compétences nous sommes confiantes de notre capacité de résoudre des problèmes futurs liés à l'électronique et le ML.

---

## BIBLIOGRAPHIE

- [1] <https://www.ompe.org/author/julia>. Causes et conséquences des incendies de forêt, June 2017.
- [2] L'Économiste. Incendies de forêts au maroc : plus de 22 700 ha touchés en 2022. *L'Économiste*, 2022.
- [3] Les incendies de forêt | IFRC.
- [4] European Commission. Joint Research Centre. *Advance report on wildfires in Europe, Middle East and North Africa 2021*. Publications Office, LU, 2022.
- [5] effis. Annual report 2021. URL : [https://effis-gwis-cms.s3.eu-west-1.amazonaws.com/effis/reports-and-publications/annual-fire-reports/Annual\\_Report\\_2021\\_final\\_topdf1.pdf](https://effis-gwis-cms.s3.eu-west-1.amazonaws.com/effis/reports-and-publications/annual-fire-reports/Annual_Report_2021_final_topdf1.pdf), 2021. Accessed : 10/04/2023.
- [6] aravind-selvam/forest-fire-prediction : Project for Predicting Algerian Forest Fires and Fire Weather Index Using Machine Learning with Python.
- [7] Sagnik220/Forest-Fire-Prediction.
- [8] Aalok Jha. ForestFirePrediction, November 2022. original-date : 2020-05-28T13:26:56Z.
- [9] IoT Based Forest Fire Detection System using Arduino and GSM Module.
- [10] Forest Fire Detection and Prediction Using NodeMCU with IoT.

- [11] Maria Evita, Sensius Seno, Azka Zakiyyatuddin, Abdul Muid, and Mitra Djamar. Application of UAV (Unmanned Aerial Vehicle) Photogrammetry for Forest Fire Early Detection System. *IOP Conference Series : Earth and Environmental Science*, 830 :012025, September 2021.
- [12] Nikolay Baranovskiy. Deterministic-Probabilistic Approach to Predict Lightning-Caused Forest Fires in Mounting Areas. *Forecasting*, 3(4) :695–715, December 2021. Number : 4 Publisher : Multidisciplinary Digital Publishing Institute.
- [13] Amrita Pathak. Cycle de vie du développement logiciel (SDLC) : un guide complet, June 2022.
- [14] Futura. Définition | internet des objets - ido - internet of things - iot | futura tech. Accessed : 05/05/2023.
- [15] Synox Marketing. 4 choses à savoir sur l'Internet des Objets, May 2020.
- [16] Differences Between AI vs. Machine Learning vs. Deep Learning | Simplilearn.
- [17] Guillaume Saint-Cirgue. Apprendre machine learning en une semaine, 2019. Accessed : 14/04/2023.
- [18] Zakariyaa ISMAILI. Apprentissage Supervisé Vs. Non Supervisé, January 2019.
- [19] Guillaume Saint-Cirgue. Comment fonctionne le Machine Learning ?, March 2019.
- [20] Hugo Welter. Deep learning : définition, fonctionnement et applications.
- [21] Pololu - Arduino Uno R3.
- [22] espressif. Esp32wroom32 datasheet. URL : [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf). Accessed : 01/05/2023.
- [23] DHT11 capteur de température et humidité avec Arduino - Moussasoft, January 2023. Running Time : 458 Section : Arduino.
- [24] Mouser Electronics. DHT11 technical data sheet. Accessed : 04/05/2023.
- [25] Nathalie Mayer. Définition | Anémomètre | Futura Planète.
- [26] Espacio de Cesar. Como medir el viento muy fácil. Anemometro simple y con ARDUINO, March 2017.

- [27] Projet EU. I2C bus : Inter-integrated circuit. Accessed : 07/05/2023.
- [28] Breadboard - an overview | ScienceDirect Topics.
- [29] Joaquín García. How to install Arduino IDE on the latest Ubuntu versions, May 2018.
- [30] MathWorks. ThingSpeak Documentation. Accessed : 05/05/2023.
- [31] Python (in Machine Learning).
- [32] 1.10. Decision Trees. Accessed : 28/05/2023.
- [33] Introduction to Random Forest in Machine Learning.

---

## ANNEXE

- Code d'Arduino

```
1 //Les bibliotheques//  
2 #include <DHT.h> //la bibliotheque du capteur de temperature et humidite  
3 #include<Wire.h> // La bibliotheque de I2c  
4 #include <LiquidCrystal_I2C.h> // la bibliotheque de l'afficheur LCD  
5 LiquidCrystal_I2C lcd(0x27,16,2); //creation d'une instance de l'objet lcd  
pour utiliser l'afficheur LCD  
6 DHT dht(8, DHT11); //creation d'une instance de l'objet dht pour utiliser  
le capteur DHT11 connecte a la broche 8  
7 //Declaration des variables//  
8 float temp;  
9 float humidity;  
10 long num_v_n;  
11 float tension_asser;  
12 int v1=A3;  
13 float vent=0;  
14 //Initialisation//  
15 void setup(){  
16 Serial.begin(9600); // la vitesse de transmission  
17 pinMode(v1,INPUT); //configurer la broche v1 en mode entree  
18 dht.begin(); // initialiser le capteur DHT11  
19 lcd.backlight(); //allumer la retroclairage de l'afficheur LCD  
20 lcd.begin(16, 2); //initialiser l'afficheur LCD  
21 analogReference(INTERNAL); //configure la reference de tension analogique  
sur la tension interne de l'Arduino  
22 }  
23 //La fonction de repitition//  
24 void loop(){  
25 num_v_n=analogRead(v1); //lire la valeur numerique//
```

```
26 tension_asser=5*num_v_n/1024; //Convertir la valeur numerique en
27   analogique//
28 tension_asser=tension_asser*1.2*1000; //1.2 c'est le rapport du max du
29   moteur(6V) et max d'Arduino(5V) , et convertir la tension en milli
30   volts(x1000)//
31
32 //DHT11//
33 //Affichage de la temperature sur LCD//
34 temp=dht.readTemperature();
35 humidity=dht.readHumidity();
36 lcd.setCursor(0,0);
37 lcd.print("T: ");
38 lcd.print(temp);
39 lcd.print("C");
40 //Affichage de l'humidite sur LCD//
41 lcd.setCursor(9,0);
42 lcd.print("H: ");
43 lcd.print(humidity);
44 lcd.print("%");
45
46 //Anemometre//
47 vent=abs((tension_asser*0.190)); // la relation entre la vitesse du vent
48   et la tension
49 //Affichage de vitesse du vent sur LCD//
50 lcd.setCursor(0,1);
51 lcd.print("vent: ");
52 lcd.print(vent);
53 lcd.print("km/h");
54
55 if(isnan(temp) || isnan(humidity) || isnan(vent))
56 {
57   Serial.println("No data available");
58 }
59
60 Serial.print(temp); //Affichage de la temperature sur moniteur serie//
61 Serial.print(",");
62 Serial.print(humidity); //Affichage de l'humidite sur moniteur serie//
63 Serial.print(";");
64 Serial.println(vent); //Affichage de la vitesse du vent sur moniteur serie
65   //
66 delay(1000); //une pause de 1000 millisecondes ( 1 seconde ) avant de
67   recommencer
68 }
```

- Code d'ESP-32 et ThingSpeak

```
1 // les broches utilisees pour la transmission serie des donnees//  
2 #define RX 16 // la broche de ESP-32 qui recoie les donnees  
3 #define TX 17 // la broche de ESP-32 qui envoie les donnees  
4 //les bibliotheques//  
5 #include <WiFi.h> // la bibliotheque WiFi qui permet la gestion de la  
    connexion WiFi  
6 #include<ThingSpeak.h> //la bibliotheque ThingSpeak , qui permet l'envoi  
    des donnees a ThingSpeak  
7 WiFiClient client;// creation de l'objet "client" de la classe "  
    WiFiClient"  
8 unsigned long mychannelnumber= ID;//le numero de canal sur ThingSpeak  
9 const char* apiKey = "apikey";//la cle API de ThingSpeak  
10 char ssid [] = "ssid"; //nom du reseau WiFi  
11 char pass [] = "password";//mot de passe du reseau WiFi  
12 //declaration des variable//  
13 float t;  
14 float h;  
15 float v;  
16 void setup() {  
17 Serial.begin(115200);// la communication serie a une vitesse de 115200  
    bauds  
18 Serial2.begin(9600, SERIAL_8N1, RX, TX); //communication serie sur la  
    deuxieme interface serie a une vitesse de 9600 bauds  
19 delay(10);  
20 Serial.println("Connecting to ");  
21 Serial.println(ssid);  
22 WiFi.begin(ssid , pass); //initialisation de la connexion WiFi  
23 while (WiFi.status() != WL_CONNECTED)  
24 {  
25     delay(500);  
26     Serial.print(".");  
27 }  
28 Serial.println("");  
29 Serial.println("connected");  
30 ThingSpeak.begin(client); // initialisation de la connexion avec le  
    serveur ThingSpeak en utilisant l'objet client  
31 }  
32 void loop()  
33 {  
34 String donnees=Serial2.readString(); // lire et stocker les donnees recues  
    sur l'interface serie 2  
35 Serial.println(donnees);  
36 int comma=donnees.indexOf( ',' );
```

```
37 int commadot=donnees.indexOf(';');
38 if(comma!=-1 && commadot!=-1)//verifie si les indices des virgules et des
   points-virgules ont ete trouves
39 {
40     t=donnees.substring(0,comma).toFloat();// convertir et stocker cette
       partie en nombre a virgule flottante
41     h=donnees.substring(comma+1,commadot).toFloat();
42     v=donnees.substring(commadot+1).toFloat();
43 }
44 //Affichage sur moniteur serie//
45 Serial.print("Temperature: ");
46 Serial.println(t);
47 Serial.print("humidity: ");
48 Serial.println(h);
49 Serial.print("vent: ");
50 Serial.println(v);
51
52 ThingSpeak.setField(1,t);//champ 1 sur ThingSpeak avec la valeur actuelle
   de la variable t, qui represente la temperature
53 ThingSpeak.setField(2,h);//champ 2 sur ThingSpeak avec la valeur actuelle
   de la variable h, qui represente l'humidite
54 ThingSpeak.setField(3,v);//champ 3 sur ThingSpeak avec la valeur actuelle
   de la variable v, qui represente la vitesse du vent
55 ThingSpeak.writeFields(mychannelnumber, apiKey);//envoyer les valeurs des
   champs definis precedemment a ThingSpeak
56 delay(1000); //une pause de 1000 millisecondes (1 seconde) avant de
   recommencer
57 }
```