

# Table des matières

<b>Dédicaces</b>	<b>i</b>
<b>Remerciements</b>	<b>i</b>
<b>Table des figures</b>	<b>iii</b>
<b>Liste des tableaux</b>	<b>iv</b>
<b>Liste des abréviations</b>	<b>x</b>
<b>Introduction générale</b>	<b>x</b>
<b>1 Cadre Général du Projet</b>	<b>1</b>
1 Présentation de l'organisme d'accueil . . . . .	2
2 Cadre du projet . . . . .	2
2.1 La problématique . . . . .	2
2.2 Objectifs majeurs du projet . . . . .	5
2.3 Description et critique de l'existant . . . . .	6
2.4 Solution proposée . . . . .	9
3 Méthodologie utilisée . . . . .	9
3.1 Méthode de gestion de projet agile Scrum . . . . .	10
3.2 Méthode de modélisation et de conception . . . . .	11
<b>2 Analyse et spécification des besoins</b>	<b>13</b>
1 Spécification des besoins . . . . .	14
1.1 Diagramme des exigences . . . . .	14
1.2 Modélisation du besoin . . . . .	16
2 Pilotage du projet avec scrum . . . . .	18
2.1 Équipe et rôle . . . . .	18
2.2 Backlog de Produit . . . . .	18
2.3 Diagramme de Gantt . . . . .	19
<b>3 Elaboration des modèles IA</b>	<b>22</b>
1 Le modèle de détection de feu au niveau de la caméra . . . . .	23
1.1 Les tests des modèles de Détection d'Incendie . . . . .	24
1.2 Comparaison des Modèles TensorFlow Lite . . . . .	25
1.3 Étapes de création d'un modèle personnalisé de détection de feu . . . . .	27
2 Le Modèle de détection d'odeurs au niveau de BME688 . . . . .	27
2.1 la présentation du capteur : . . . . .	27

## TABLE DES MATIÈRES

---

2.2	les étapes de développement de modèle de détection : . . . . .	28
<b>Résumé</b>		<b>31</b>

# Table des figures

Figure 1.1	Logo commercial du Smart For Green [1]	2
Figure 1.2	émission annuelle du gaz CO <sub>2</sub> dans le monde et dans la tunisie	3
Figure 1.3	le cycle d'émission de carbone et d'absorption naturelle des équivalents carbone en gigatonnes	4
Figure 1.4	Incendies et boucle de rétroaction climatique	5
Figure 1.5	Processus général du projet	6
Figure 1.6	Exemple d'image satellite utilisée pour la détection des incendies	7
Figure 1.7	Exemple de capteur au sol utilisé pour la détection des incendies	7
Figure 1.8	Exemple de drone utilisé pour la détection des incendies	8
Figure 1.9	Cycle de vie de la méthodologie Scrum[2]	10
Figure 1.10	Logo UML	11
Figure 1.11	Logo SysML	12
Figure 2.1	Diagramme d'exigence de système	15
Figure 2.2	Diagramme de cas d'utilisation général du système de surveillance	17
Figure 2.3	Présentation de l'équipe SCRUM	18
Figure 3.1	Résultats des essais du modèle FOMO sur la plateforme Edge Impulse	24
Figure 3.2	Résultat visuel du modèle YOLO sur la plateforme RoboFlow	25
Figure 3.3	le capteur BME688	28
Figure 3.4	Collecte des Données avec le Raspberry Pi 4	29
Figure 3.5	Préparation des Données	29
Figure 3.6	Évaluation des Modèles	30
Figure 3.7	Enregistrement et Déploiement du Modèle	30

# Liste des tableaux

1.1	Comparaison des solutions de détection des incendies . . . . .	8
2.1	Description des acteurs . . . . .	16
2.2	Backlog de Produit pour le Projet IoT de Détection d'Incendie . . . . .	20
2.3	Diagramme de Gantt Simplifié pour le déroulement de ce projet . . . . .	21
3.1	Comparaison de MobileNet, FOMO et YOLO pour la détection d'incendie . . .	23
3.2	Comparaison des Modèles TensorFlow Lite pour la Détection d'Objets . . . .	25
3.3	Comparaison Générale des Algorithmes . . . . .	30

# Liste des Abréviations

<b>ABP</b>	: Activation By Personalization
<b>BUI</b>	: Buildup Index
<b>DC</b>	: Drought Code
<b>DMC</b>	: Duff Moisture Code
<b>FFMC</b>	: Fire Fuel Moisture Code
<b>FWI</b>	: Fire Weather Index
<b>IOT</b>	: Internet of Things
<b>ISI</b>	: Initial Spread Index
<b>IT</b>	: Information Technology
<b>LoRaWAN</b>	: Long Range Wide Area Network
<b>ML</b>	: Machine learning
<b>MQTT</b>	: Message Queuing Telemetry Transport
<b>OTAA</b>	: Over-The-Air Activation

# Introduction générale

L'évolution rapide de l'Internet des Objets (IoT) et l'essor de l'intelligence artificielle (IA) ont entraîné des transformations majeures dans de nombreux secteurs, y compris l'agriculture intelligente. Grâce aux dispositifs connectés, les agriculteurs peuvent désormais accéder en temps réel à des données sur les conditions agricoles, leur permettant ainsi de prendre des décisions plus informées pour une gestion optimale de leurs cultures.

Dans un monde où les risques d'incendies dans les fermes agricoles sont exacerbés par les conditions météorologiques changeantes, la nécessité d'une surveillance et d'une prévention efficaces devient impérative. C'est dans ce contexte que notre projet de fin d'études, intitulé "Conception et réalisation d'un système IoT de détection d'incendie dans les zones agricoles", a pris forme. Mené au sein de notre cursus de Master professionnel Robotique avancée et intelligence artificielle à l'Institut Supérieur des Études Technologiques de Bizerte, en collaboration avec la startup Smart For Green.

Ce projet vise à combiner les technologies de l'Internet des objets (IoT) et de l'intelligence artificielle (IA) pour renforcer la sécurité des exploitations agricoles. En intégrant la collecte de données en temps réel via l'IoT et l'analyse avancée des données par l'IA, notre objectif est d'instaurer une détection précoce des incendies, offrant ainsi une protection accrue contre ce type de catastrophe dans les milieux agricoles.

Pour atteindre cet objectif, ce rapport détaille le travail accompli à travers plusieurs chapitres :

- Le premier chapitre, intitulé "**Cadre Général du projet**", présente l'organisme d'accueil, décrit le contexte de notre projet ainsi que la méthodologie adoptée.
- Le deuxième chapitre, "**Analyse et spécification des besoins**", explique notre démarche, notamment l'identification des acteurs du système et l'analyse des besoins.
- Le troisième chapitre, "**Élaboration des Modèles IA**", couvre la collecte et la préparation des données, ainsi que la conception, l'entraînement et la validation des modèles de machine learning.
- Le quatrième chapitre, "**Développement de l'Application Web**", décrit la conception de l'interface utilisateur, ainsi que le développement du frontend et du backend de l'application.
- Le cinquième chapitre, "**Réalisation du Système IoT**", détaille l'architecture matérielle, la configuration et les tests de la communication LoRa, ainsi que l'intégration du matériel avec le logiciel.

Enfin, ce rapport se conclut par une synthèse générale et une discussion sur les perspectives futures de ce travail.

## CADRE GÉNÉRAL DU PROJET

---

## Introduction

Ce premier chapitre sera consacré à la présentation générale du projet. En effet, nous décrivons en premier lieu l'organisme d'accueil où nous avons effectué notre stage, ensuite nous allons décrire le cadre général de notre projet, enfin nous abordons l'étude de l'existant et la solution envisagée.

## 1 Présentation de l'organisme d'accueil

**Smart for Green** est une Startup créée en 2022 au sein du centre local d'innovations de Ras Jebel-Bizerte, qui a principalement été un bureau d'études qui offre une variété de services liés à l'agriculture intelligente, tels que la prévention des feux et la protection contre les incendies de forêt. La startup Smart for Green est dirigée par une équipe une talentueuse et déterminée de professionnels passionnés par leur mission dirige la startup Smart for Green. Chaque individu apporte sa propre expertise et sa passion débordante à l'entreprise, et je suis enchanté de collaborer pour réaliser une vision commune.

Le logo commercial de la startup est représenté dans la figure 1.1.



FIGURE 1.1 – Logo commercial du Smart For Green [1]

## 2 Cadre du projet

### 2.1 La problématique

#### 2.1.1 La problématique de la concentration de CO<sub>2</sub> dans l'air dans le monde et en Tunisie

La concentration de dioxyde de carbone (CO<sub>2</sub>) dans l'atmosphère est un problème mondial croissant qui a des conséquences graves sur l'environnement. Le CO<sub>2</sub>, principal gaz à effet de serre, provient principalement de la combustion des combustibles fossiles et de la déforestation. Depuis l'ère préindustrielle, les niveaux de CO<sub>2</sub> ont augmenté de manière exponentielle, atteignant des concentrations sans précédent dans l'histoire humaine. En Tunisie, les niveaux de

CO<sub>2</sub> augmentent également, en raison de l'industrialisation et de l'urbanisation rapides. **GFED** La figure 1.2 montre l'augmentation annuelle d'émission du gaz de dioxyde de carbone dans le monde et dans la Tunisie.

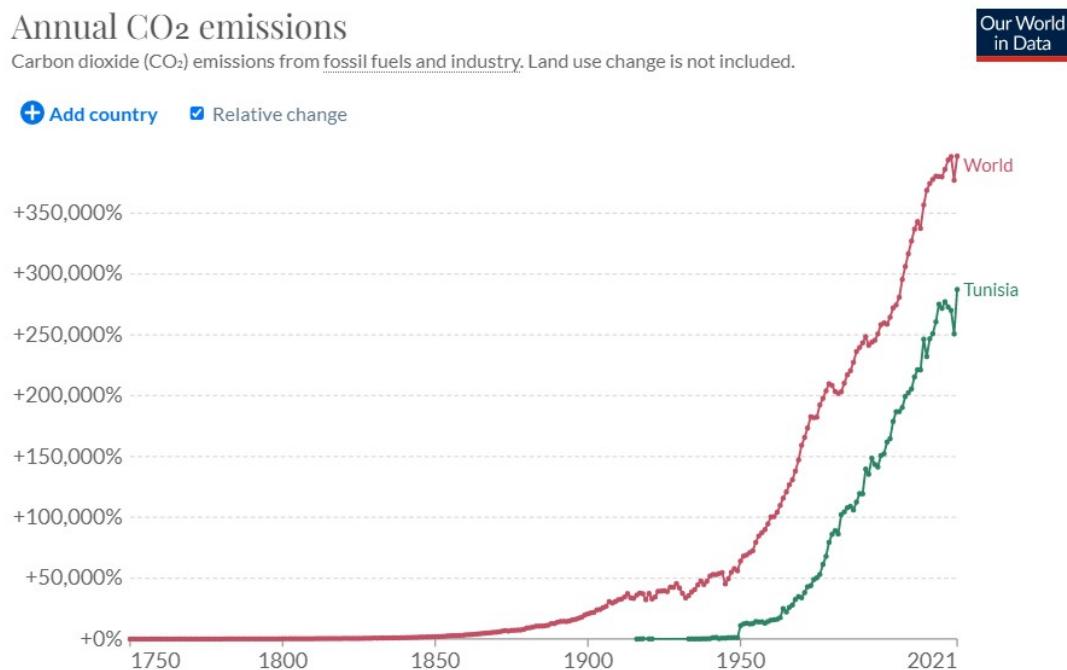


FIGURE 1.2 – émission annuelle du gaz CO<sub>2</sub> dans le monde et dans la tunisie

### 2.1.2 L'effet de serre et le trou de l'ozone

L'effet de serre est un phénomène naturel où certains gaz dans l'atmosphère terrestre retiennent la chaleur du soleil, réchauffant ainsi la planète. Cependant, l'activité humaine a intensifié cet effet, en augmentant les niveaux de gaz à effet de serre comme le CO<sub>2</sub>, le méthane et les oxydes d'azote. Cette augmentation entraîne une hausse des températures globales, perturbant les écosystèmes et les cycles climatiques.

Ainsi, la couverture végétale est le facteur le plus important dans le cycle du carbone, comme le montre la figure 1.3.

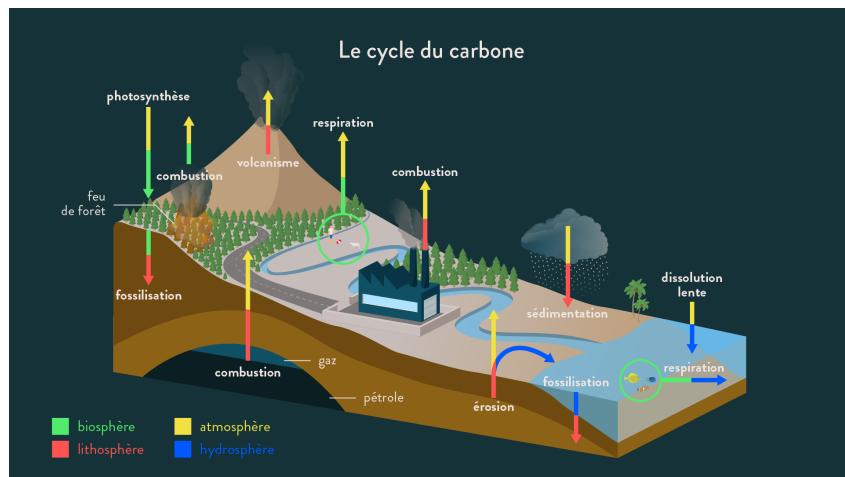


FIGURE 1.3 – le cycle d'émission de carbone et d'absorption naturelle des équivalents carbone en gigatonnes

Le trou dans la couche d'ozone est un autre problème environnemental majeur. Bien que le CO<sub>2</sub> ne soit pas directement responsable du trou d'ozone, d'autres gaz à effet de serre, comme les chlorofluorocarbures (CFC), ont contribué à l'amincissement de cette couche protectrice. La couche d'ozone joue un rôle crucial en absorbant la majorité des rayons ultraviolets nocifs du soleil. Un amincissement de cette couche entraîne une augmentation de l'exposition aux UV, ce qui a des effets négatifs sur la santé humaine, les écosystèmes marins et terrestres, et les cultures agricoles. **NASA**

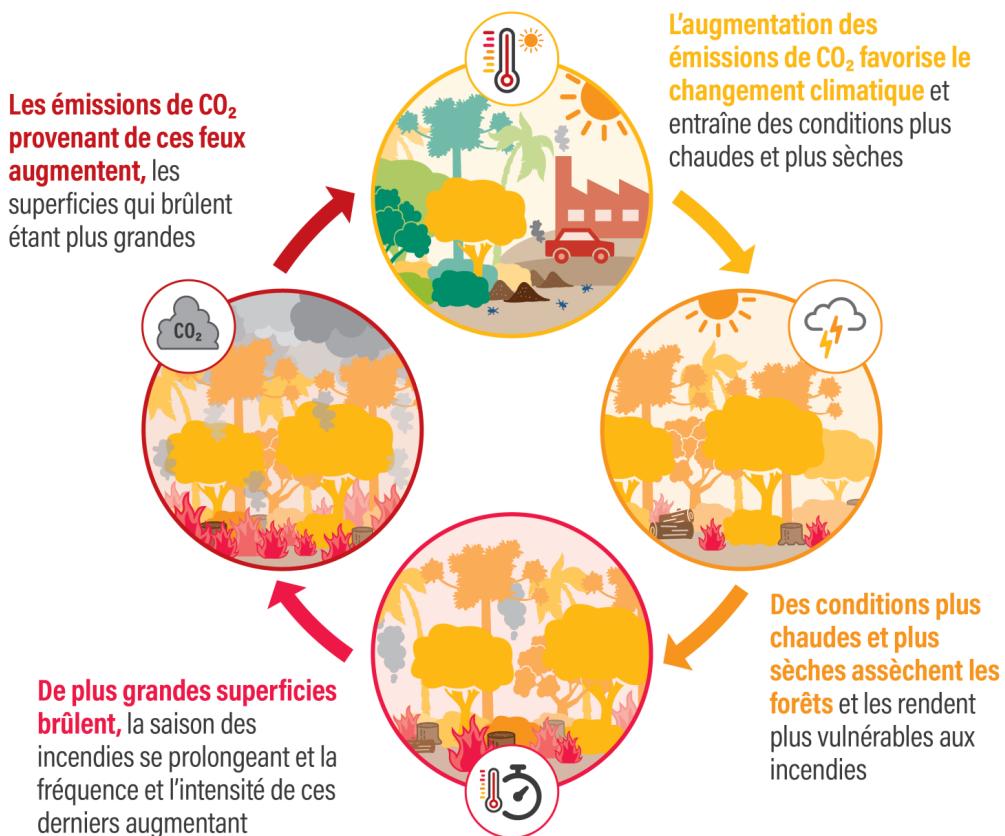
### 2.1.3 Conséquences négatives sur l'environnement et les zones agricoles

L'augmentation des niveaux de CO<sub>2</sub> et d'autres gaz à effet de serre a plusieurs conséquences néfastes :

1. **Changement climatique** : Les températures globales en hausse entraînent des phénomènes météorologiques extrêmes, tels que des vagues de chaleur, des sécheresses, des inondations et des tempêtes plus fréquentes et plus intenses.
2. **Dégradation des terres agricoles** : Les conditions climatiques extrêmes affectent la productivité des cultures. En Tunisie, les sécheresses prolongées et les variations climatiques perturbent les cycles de croissance des plantes, réduisant les rendements agricoles.
3. **Biodiversité en danger** : Les écosystèmes terrestres et marins sont perturbés par le changement climatique, menaçant de nombreuses espèces animales et végétales.

### 2.1.4 Nombre des incendies par an dans le monde et en Tunisie

Les incendies de forêt sont de plus en plus fréquents et destructeurs en raison du changement climatique. Selon les données de la Global Fire Emissions Database (GFED), il y a eu environ 350 000 incendies de forêt dans le monde en 2020. En Tunisie, les incendies de forêt sont également en augmentation. Le Ministère de l'Agriculture, des Ressources Hydrauliques et de la Pêche enregistre en moyenne 100 à 150 incendies de forêt par an, affectant des milliers d'hectares de terres.[2]



Source: Global Forest Watch.  
22.08.01



WORLD RESOURCES INSTITUTE

FIGURE 1.4 – Incendies et boucle de rétroaction climatique

## 2.2 Objectifs majeurs du projet

L'objectif de notre projet est de concevoir et de développer un système de surveillance des incendies de forêt basé sur la prédiction de l'emplacement de l'incendie grâce à un noeud de capteur. Un feu est détecté par un appareil photo intelligent qui fonctionne avec la reconnaissance d'images grâce à l'apprentissage automatique (Tiny ML). Il permet de signaler rapidement le début d'incendie et d'intervenir rapidement les équipes de sauvetage afin de limiter sa propagation, comme indiqué dans la figure 1.4.

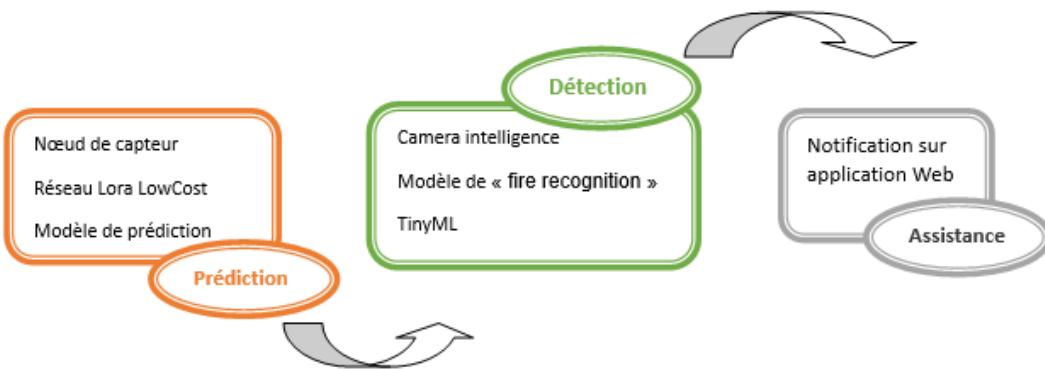


FIGURE 1.5 – Processus général du projet

Pour atteindre notre objectif, nous étudions en premier lieu la direction de la surveillance des incendies des forêts dans le monde entier et en particulier, en Tunisie. Ceci nous permettra d'identifier les acteurs de notre application. En second lieu, nous passons à une étude de quelques technologies utilisés pour choisir la technique adaptée à notre solution, ainsi que le matériel nécessaire à configurer. Ensuite, nous faisons une étude conceptuelle afin de dégager les différentes structures et scénarios de notre solution.

## 2.3 Description et critique de l'existant

Les incendies dans les zones agricoles et forestières représentent un grave danger pour l'environnement, les écosystèmes et les économies locales. La détection précoce des incendies est cruciale pour minimiser les dégâts et permettre une intervention rapide des services de secours. Dans cette section, nous explorerons les principales solutions existantes pour la détection des incendies dans ces zones sensibles.

### 2.3.1 Solutions existantes pour la détection des incendies

**Systèmes de détection par satellite** Les systèmes de détection par satellite utilisent des images satellites pour identifier les zones de chaleur anormales qui pourraient indiquer un incendie. Ces systèmes sont capables de couvrir de vastes étendues de terrain et de fournir des données en temps quasi réel.

- **Avantages** : Large couverture géographique, capacité à surveiller des zones inaccessibles, détection rapide sur une vaste échelle.
- **Inconvénients** : Résolution limitée des images, dépendance aux conditions météorologiques, coûts élevés de mise en œuvre et de maintenance.



FIGURE 1.6 – Exemple d'image satellite utilisée pour la détection des incendies

**Systèmes de détection basés sur les capteurs au sol** Les systèmes de détection au sol reposent sur un réseau de capteurs placés stratégiquement dans les zones à risque. Ces capteurs peuvent détecter la fumée, la chaleur ou les gaz émis par les incendies.

- **Avantages :** Haute précision de détection, possibilité de détection locale et immédiate, coûts relativement bas pour les capteurs individuels.
- **Inconvénients :** Nécessité d'un nombre élevé de capteurs pour une couverture complète, entretien et remplacement réguliers des capteurs, dépendance à la connectivité réseau.



FIGURE 1.7 – Exemple de capteur au sol utilisé pour la détection des incendies

**Systèmes de détection par drones** Les drones équipés de caméras thermiques et de capteurs peuvent survoler les zones à risque pour détecter les signes d'incendie. Ils offrent une flexibilité

et une rapidité d'intervention importantes.

- **Avantages** : Mobilité et flexibilité, capacité à accéder à des zones difficiles d'accès, collecte de données en temps réel.
- **Inconvénients** : Autonomie limitée des drones, besoin de pilotes qualifiés, coûts d'acquisition et d'entretien élevés.



FIGURE 1.8 – Exemple de drone utilisé pour la détection des incendies

### 2.3.2 Critique des solutions existantes

Les trois principales solutions de détection des incendies présentent des avantages et des inconvénients spécifiques. Les systèmes de détection par satellite offrent une large couverture mais sont limités par la résolution des images et les conditions météorologiques. Les systèmes de détection au sol sont précis mais nécessitent un grand nombre de capteurs pour une couverture complète. Les drones, quant à eux, offrent une grande flexibilité mais sont contraints par leur autonomie et leur coût.

TABLE 1.1 – Comparaison des solutions de détection des incendies

Solution	Avantages	Inconvénients
Détection par satellite	Large couverture, détection rapide	Résolution limitée, dépendance météorologique, coûts élevés
Détection par capteurs au sol	Haute précision, détection immédiate	Besoin de nombreux capteurs, maintenance régulière
Détection par drones	Mobilité, accès à des zones difficiles	Autonomie limitée, coûts élevés, besoin de pilotes

En conclusion, bien que chaque solution présente des atouts, aucune n'est parfaite. L'intégration de plusieurs technologies pourrait offrir une meilleure couverture et une détection plus fiable des incendies dans les zones agricoles et forestières.

## 2.4 Solution proposée

La solution proposée consiste à concevoir un détecteur d'incendie IoT dans des fermes agricoles qui peut être extrêmement utile pour prévenir et détecter les incendies. Ces détecteurs peuvent être montés sur un poteau ou un arbre de 3 mètres de haut et installés à 60 mètres l'un de l'autre pour former une clôture virtuelle. L'espacement de 60 mètres garantit qu'aucun feu ne se déclenche sans être détecté.

Il permet de :

- **Identifier un feu dans un milieu agricole**

Le système de surveillance des incendies à distance proposé utilise des techniques d'intelligence artificielle et de vision par caméra pour détecter efficacement la présence de feu dans les environnements agricoles. En plus de détecter les flammes, il surveille également les paramètres environnementaux tels que la température, le taux d'humidité et les niveaux de gaz, ce qui permet une détection précoce des départs de feu.

Pour répondre aux besoins des zones éloignées, la conception matérielle du système est optimisée pour une faible consommation d'énergie et une longue autonomie de la batterie. Cela garantit que le système peut fonctionner de manière autonome pendant de longues périodes sans besoin d'entretien fréquent, ce qui est crucial dans les environnements agricoles où les ressources peuvent être limitées.

- **Envoyer une alerte**

Le système de surveillance des incendies à distance collecte en continu des données environnementales et visuelles, et en cas de détection d'incendie, il déclenche automatiquement une alerte pour signaler le danger aux personnes concernées via une application web dédiée. Grâce à un protocole de communication à longue portée, les données sont transférées sur Internet même dans les zones où la connectivité est limitée, garantissant ainsi une transmission fiable des alertes.

De plus, le système peut être configuré pour fournir un suivi en temps réel de la scène de l'incendie, ce qui permet une surveillance continue et une évaluation de la progression de l'incendie.

- **Signaler l'emplacement de la zone du feu**

Le module de détection d'incendie à distance peut également fournir des données sur l'emplacement précis de l'incendie en utilisant des coordonnées géographiques. Ces informations peuvent être visualisées sur une carte géospatiale intégrée à une interface web, permettant aux utilisateurs de localiser rapidement et avec précision l'incendie détecté.

## 3 Méthodologie utilisée

Les méthodes traditionnelles de gestion de projets se basent sur une spécification initiale très précise des besoins, leur réalisation, puis le rendu final, exige que chaque participant à la réalisation du projet termine toutes ses tâches avant de les transférer au suivant.

### 3.1 Méthode de gestion de projet agile Scrum

La méthode de gestion de projet Agile Scrum est un cadre de travail itératif et incrémental utilisé pour superviser les projets de développement de logiciels et de produits.

Elle est basée sur des principes agiles qui mettent l'accent sur la collaboration, la flexibilité et l'adaptation au changement. La méthodologie Scrum se concentre sur l'organisation et la planification des tâches en petits morceaux appelés sprints, qui sont des périodes de temps définies pour accomplir un ensemble de tâches.

La figure ci-dessous représente le cycle de vie de la méthodologie scrum :

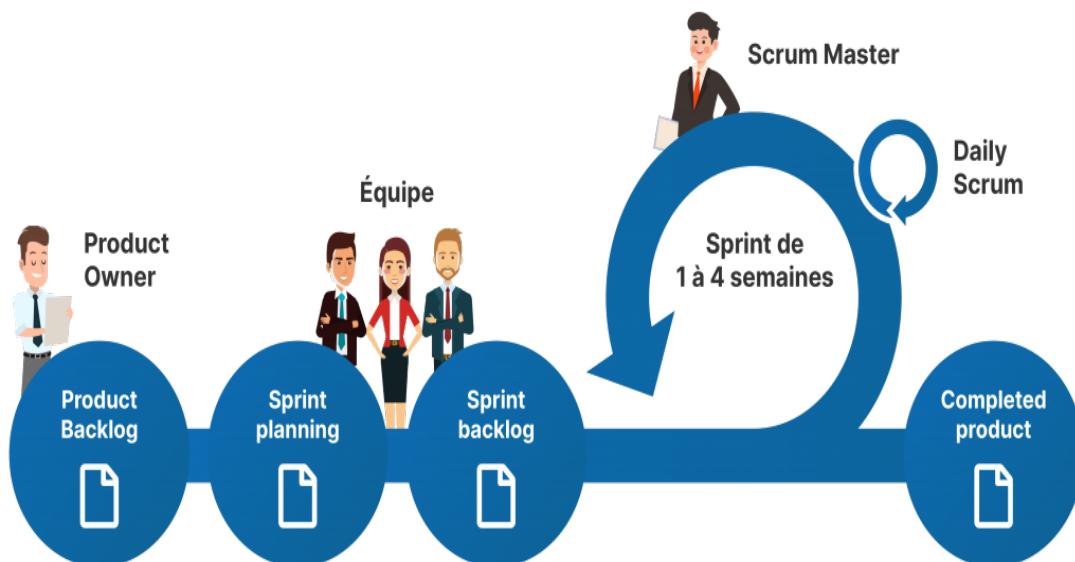


FIGURE 1.9 – Cycle de vie de la méthodologie Scrum[2]

La méthode Scrum favorise la participation active de tous les acteurs, y compris les membres de l'équipe de développement, les clients et les utilisateurs, grâce à une approche collaborative. Elle se base sur trois rôles principaux :

- ◆ **Product Owner** : est responsable de définir les objectifs et les exigences du projet
- ◆ **Scrum Master** : qui est responsable de faciliter le processus de développement
- ◆ **L'équipe de développement** : qui est responsable de la conception et de la mise en œuvre du produit.

Scrum consiste à fournir des informations complètes et détaillées plutôt que des descriptions de la manière dont tout doit être fait sur un projet, l'avancement des projets via une série d'itérations appelées sprints.

## 3.2 Méthode de modélisation et de conception

### 3.2.1 Méthode de modélisation et de conception UML

UML est un langage de modélisation graphique qui est utilisé pour représenter et concevoir des systèmes logiciels. UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet.[3]



FIGURE 1.10 – Logo UML

UML utilise des diagrammes pour représenter différents aspects d'un système.

Les diagrammes suivants sont les plus couramment utilisés :

- **Diagrammes de cas d'utilisation** : Un diagramme de cas d'utilisation aide à repérer les cas d'utilisation, les acteurs et les liens entre les cas d'utilisation et les acteurs. En fait, il représente les fonctionnalités du système du point de vue de l'utilisateur.
- **Diagrammes de classe** : Un diagramme de classe exprime de manière générale la structure statique d'un système, représenté par les classes et les relations entre les classes le plus récent.
- **Diagrammes de séquence** : Un diagramme de séquence peut représenter l'interaction d'objets en spécifiant l'ordre temporel des échanges de flux de messages dans une perspective temporelle.

### 3.2.2 Méthode de modélisation et de conception SysML

SysML est une méthode de modélisation et de conception de systèmes complexes, conçu pour répondre aux besoins croissants de modélisation de systèmes dans divers domaines les exemples incluent l'aérospatiale, l'automobile et les systèmes embarqués. SysML est une extension d'UML, une approche de modélisation pour la conception de logiciels.[4]

Voici les principales étapes de l'approche de modélisation et de conception SysML :

- Identifier les besoins du système
- Établir l'architecture du système

- Développer les exigences du système
- Modéliser le comportement du système
- Valider et vérifier le modèle



FIGURE 1.11 – Logo SysML

Les diagrammes SysML utilisés :

- **Diagramme d'exigence** : un diagramme d'exigence précise les fonctions à réaliser .
- **Diagramme d'activité** : Un diagramme d'activité est utilisé pour modéliser le flux de travail au sein d'un cas d'utilisation ou entre plusieurs cas d'utilisation, ou pour décrire la logique d'une opération.
- **Diagramme de définition de bloc** : Le diagramme de définition de bloc décrit la structure d'un système. Il permet de présenter les composants d'un système et d'identifier le flux MEI (matière, énergie, information) traversant chaque bloc dans une situation donnée.

## Conclusion

Dans ce chapitre, nous commençons par présenter l'entreprise partenaire, Smart for Green, ainsi que le contexte global de notre projet de fin d'études. Ensuite, nous introduisons la méthodologie Agile Scrum, choisie pour assurer un suivi efficace de notre projet.

Dans le chapitre suivant, nous nous attacherons à analyser et à définir les besoins de notre système de manière approfondie.

## **ANALYSE ET SPÉCIFICATION DES BESOINS**

---

## Introduction

Dans ce chapitre, nous présenterons la mise en œuvre du pilotage de projet avec Scrum. Nous commencerons par une étude préliminaire qui permettra de décrire nos besoins spécifiques. Ensuite, nous aborderons la planification nécessaire pour mener à bien le travail demandé par la création d'un diagramme de cas d'utilisation global. Enfin Nous décrirons également l'environnement de travail ainsi que la planification des sprints.

# 1 Spécification des besoins

## 1.1 Diagramme des exigences

Le diagramme ci-dessous présente la décomposition hiérarchique des exigences fonctionnelles du système de détection d'incendies en zones agricoles. L'exigence principale est de détecter les incendies et de générer des alertes en temps réel. Les sous-exigences détaillent les fonctionnalités spécifiques, telles que la localisation précise des incendies, la détection par IA, la surveillance des variables climatiques, et l'autonomie énergétique.

Les principales exigences fonctionnelles sont les suivantes :

**Détection et localisation :** Le système doit détecter les incendies et localiser leur emplacement avec précision.

**Intelligence artificielle :** L'utilisation de l'IA permettra une détection précoce des incendies.

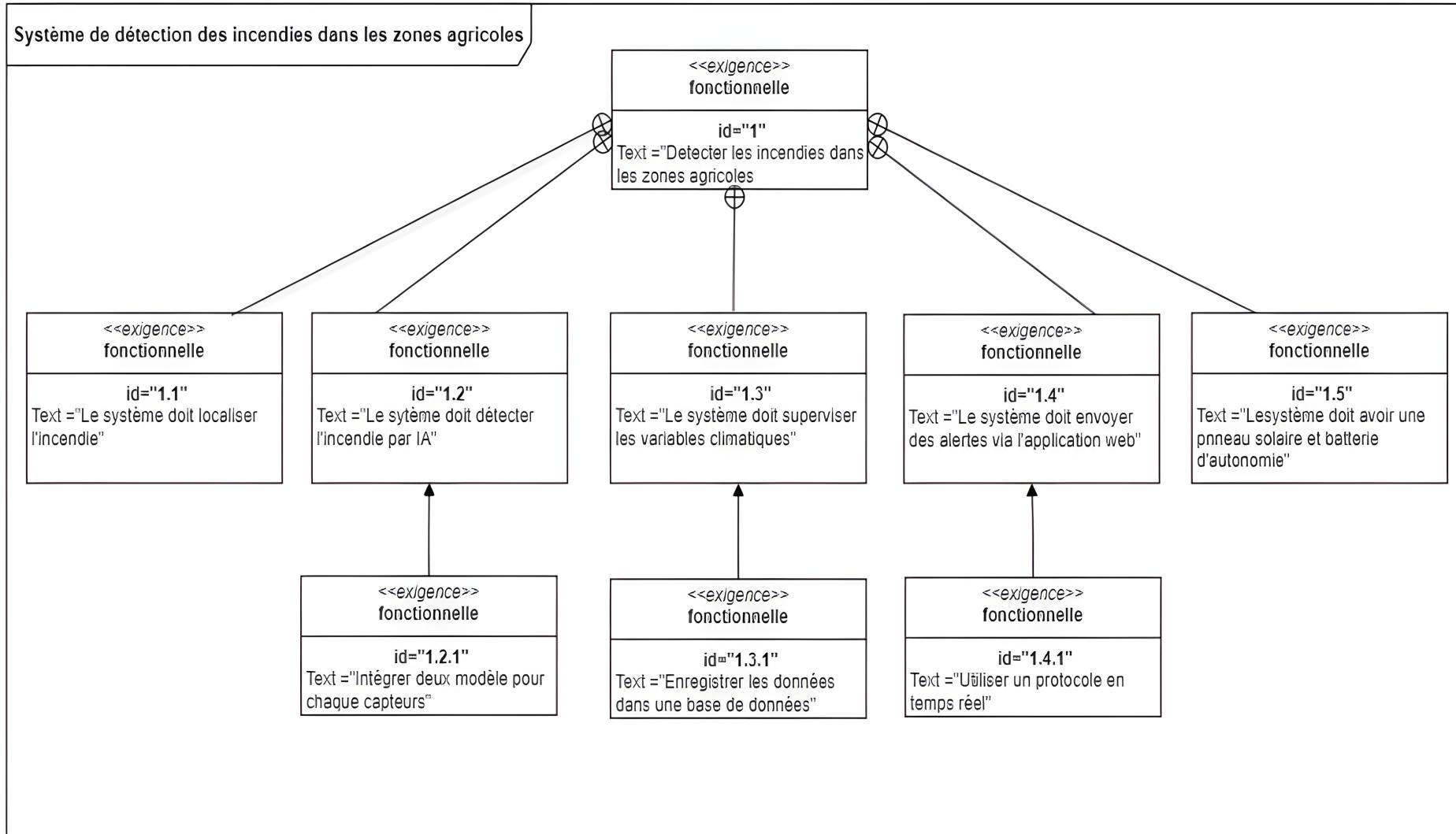
**Surveillance environnementale :** Le système surveillera les variables climatiques pour améliorer la précision de la détection.

**Alertes et communication :** Des alertes seront envoyées en temps réel via une interface web.

**Autonomie :** Le système fonctionnera grâce à une source d'énergie autonome.

**Intégration et stockage :** Le système intégrera différents capteurs et stockera les données dans une base de données.

Ce diagramme sert de base pour la conception détaillée du système et garantit que toutes les fonctionnalités essentielles sont prises en compte.



**FIGURE 2.1 – Diagramme d'exigence de système**

## 1.2 Modélisation du besoin

### 1.2.1 Identifications des acteurs

Les acteurs sont des entités externes qui interagissent avec le système. Comme indiqué ci-dessous, trois acteurs principaux sont impliqués dans notre projet.

TABLE 2.1 – Description des acteurs

Acteur	Fonctionnalités
<b>Admin</b>	<ul style="list-style-type: none"><li>★ S'authentifier</li><li>★ Gérer la liste des Superviseurs :<ul style="list-style-type: none"><li>- Ajouter un superviseur</li><li>- Supprimer un superviseur</li><li>- Modifier les informations d'un superviseur</li><li>- Afficher la liste des superviseurs</li></ul></li></ul>
<b>Superviseur</b>	<ul style="list-style-type: none"><li>★ S'authentifier</li><li>★ Gérer la liste des projets :<ul style="list-style-type: none"><li>- Ajouter un nouveau projet de client</li><li>- Supprimer un projet de client</li><li>- Consulter un projet de client</li></ul></li><li>★ Gérer les listes des clients :<ul style="list-style-type: none"><li>- Ajouter un client</li><li>- Mettre à jour les informations d'un client</li><li>- Supprimer un client</li><li>- Consulter les informations d'un client</li></ul></li><li>★ Gérer les nœuds :<ul style="list-style-type: none"><li>- Ajouter un nouveau nœud</li><li>- Supprimer un nœud</li></ul></li></ul>
<b>Client : forestier/fermier</b>	<ul style="list-style-type: none"><li>★ S'authentifier</li><li>★ Consulter le tableau de bord du projet pour accéder aux prédictions d'incendies</li><li>★ Être alerté en cas de détection d'incendie</li></ul>

### 1.2.2 Diagramme de cas d'utilisation global

Nous présentons dans la figure 2.2 le diagramme de cas d'utilisation général qui donne une vue globale du comportement fonctionnel de notre système.

Ce diagramme montre trois acteurs de même type, à savoir l'utilisateur. Tous les cas d'utilisation sont conditionnés par l'authentification, ce qui est exprimé par une relation "include" entre tous les cas d'utilisation et le cas d'utilisation "s'authentifier".

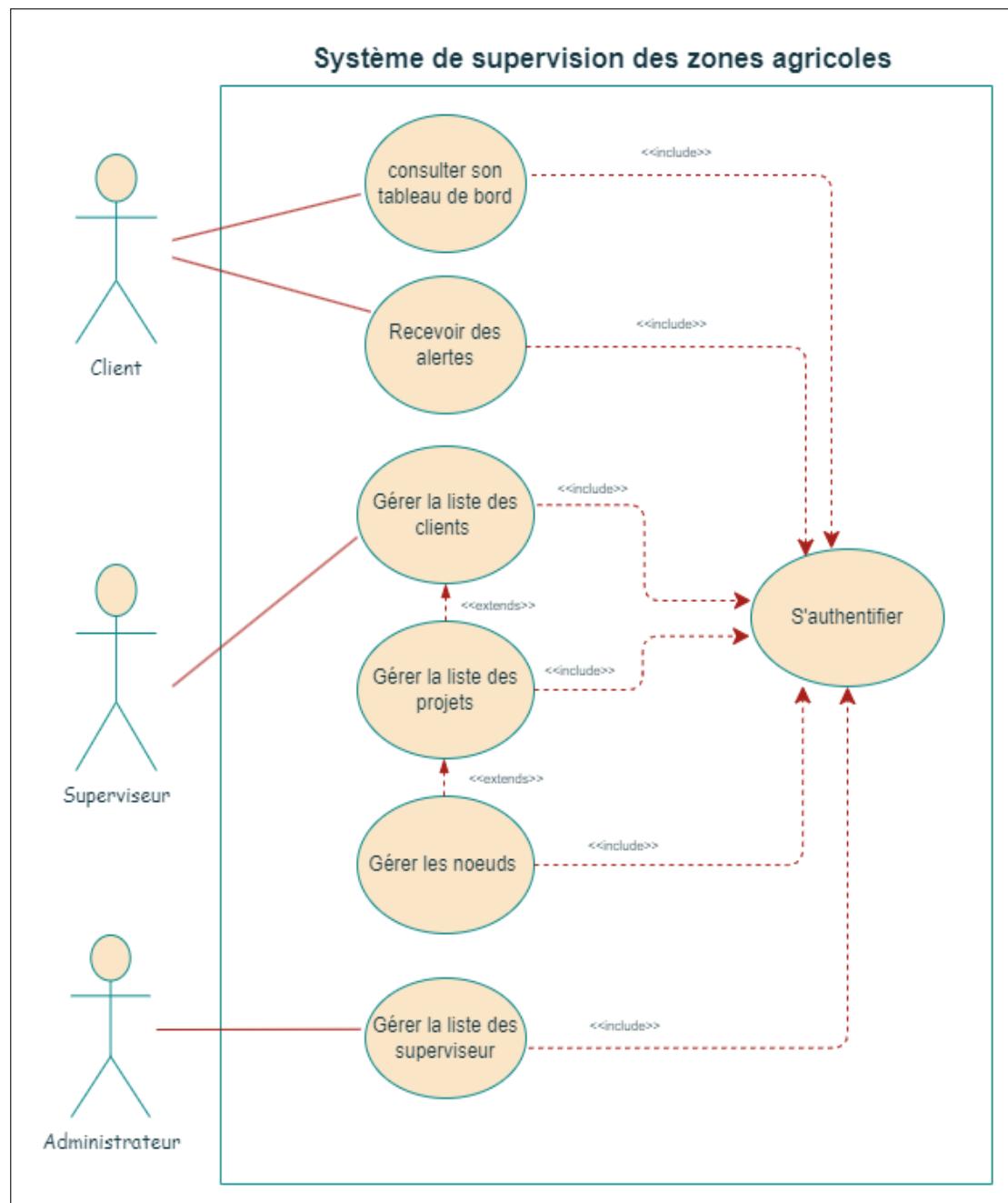


FIGURE 2.2 – Diagramme de cas d'utilisation général du système de surveillance

## 2 Pilotage du projet avec scrum

Dans le cadre de ce projet, la méthodologie Scrum a été adoptée pour structurer et organiser le travail en équipe de manière efficace et flexible. Scrum est un cadre de gestion de projet agile qui se concentre sur la livraison incrémentale du produit à travers des cycles de travail appelés sprints. L'équipe Scrum est composée de plusieurs rôles clés, chacun jouant un rôle essentiel dans la réussite du projet.

### 2.1 Équipe et rôle

Pour notre application les rôles sont répartis comme suit :

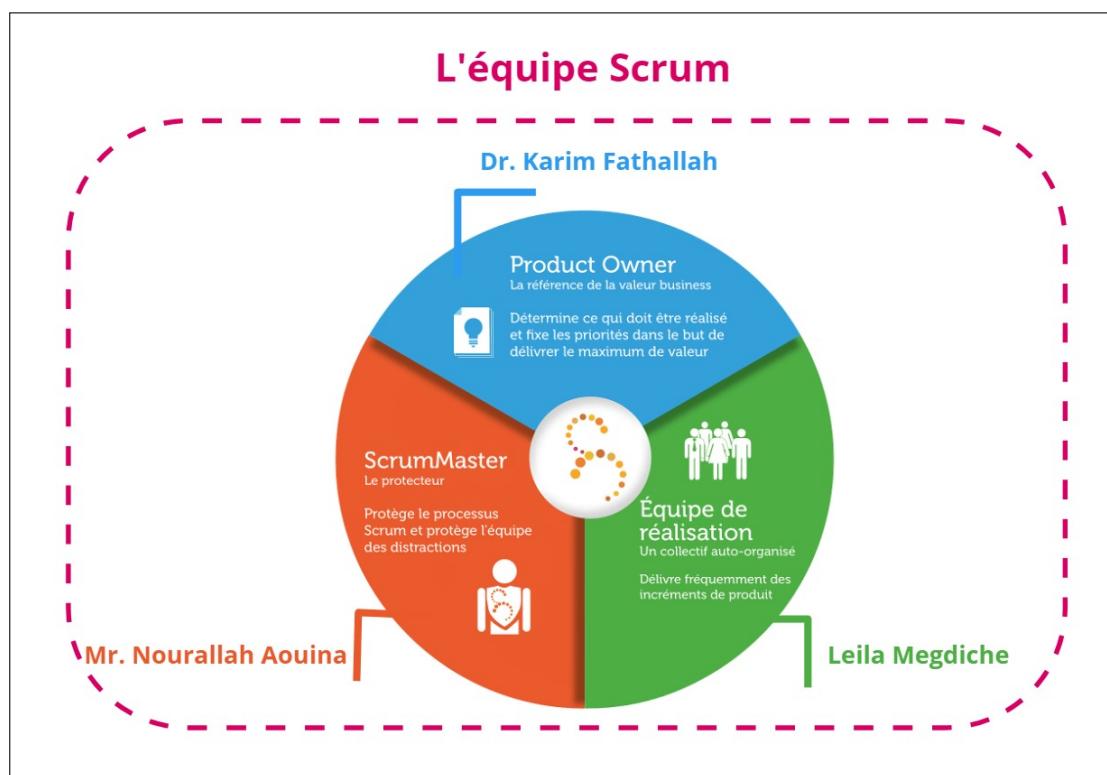


FIGURE 2.3 – Présentation de l'équipe SCRUM

### 2.2 Backlog de Produit

Le Backlog du produit joue un rôle essentiel dans le processus de Scrum. En réalité, il s'agit de toutes les fonctionnalités qui constituent le produit. Ces attributs sont désignés sous le nom d'histoires d'utilisateurs (user story). La priorité de chaque histoire utilisateur est établie par le Product Owner.

Selon notre système, le Backlog produit est le suivant :

## 2.3 Diagramme de Gantt

Le diagramme de Gantt que nous avons utilisé pour ce projet sert non seulement à visualiser l'ensemble des tâches et des phases, mais aussi à planifier et suivre l'avancement des sprints. Chaque ligne du diagramme de Gantt peut être considérée comme un sprint, où des tâches spécifiques sont assignées à l'équipe de développement. Cette approche permet une meilleure gestion du temps et des ressources, en s'assurant que chaque tâche est réalisée dans les délais impartis.

TABLE 2.2 – Backlog de Produit pour le Projet IoT de Détection d’Incendie

ID	User Story	Priorité	Critères d’acceptation
<b>1. Planification &amp; Conception</b>			
US001	Définir les objectifs du projet et les spécifications techniques	Haute	Document approuvé
US002	Réaliser une analyse des besoins du système	Haute	Document d’analyse validé
US003	Rechercher et sélectionner les technologies adaptées	Moyenne	Liste des technologies documentée
<b>2. Élaboration des Modèles IA</b>			
US004	Collecter et préparer les données nécessaires	Haute	Données prêtes pour l’entraînement
US005	Concevoir et entraîner le modèle de détection	Haute	Modèle avec précision acceptable
US006	Tester et valider les modèles IA	Moyenne	Précision d’au moins 90%
<b>3. Développement de l’Application Web</b>			
US007	Spécifier les fonctionnalités de l’application Web	Haute	Document de spécifications finalisé
US008	Concevoir l’interface utilisateur (UI/UX)	Moyenne	Maquettes UI/UX validées
US009	Développer le backend et le frontend de l’application Web	Haute	Application fonctionnelle
<b>4. Réalisation du Système IoT</b>			
US010	Choisir l’architecture matérielle et les composants	Haute	Liste des composants validée
US011	Configurer et tester la communication LoRa	Haute	Communication LoRa stable
US012	Intégrer le matériel IoT avec le logiciel	Haute	Système IoT intégré
US013	Tester et valider le système IoT complet	Moyenne	Système fonctionnel en conditions réelles
<b>5. Documentation &amp; Finalisation</b>			
US014	Rédiger le rapport final du projet	Haute	Rapport complet et approuvé
US015	Effectuer une révision et apporter les corrections	Moyenne	Rapport prêt pour la soumission
US016	Préparer la présentation pour la soutenance	Moyenne	Présentation prête

TABLE 2.3 – Diagramme de Gantt Simplifié pour le déroulement de ce projet

N°	Tâche	Début	Fin
<b>1. Planification &amp; Conception</b>			
1.1	Définition des objectifs et spécifications	Février 2024	Février 2024
1.2	Analyse des besoins	Février 2024	Février 2024
1.3	Recherche et choix des technologies	Février 2024	Mars 2024
<b>2. Élaboration des Modèles IA</b>			
2.1	Collecte et préparation des données	Mars 2024	Mars 2024
2.2	Conception et entraînement des modèles	Avril 2024	Avril 2024
2.3	Tests et validation des modèles	Avril 2024	Mai 2024
<b>3. Développement de l'Application Web</b>			
3.1	Spécifications de l'application	Avril 2024	Avril 2024
3.2	Conception UI/UX	Mai 2024	Juin 2024
3.3	Développement backend et frontend	Mai 2024	Juin 2024
<b>4. Réalisation du Système IoT</b>			
4.1	Architecture matérielle et choix des composants	Mai 2024	Juin 2024
4.2	Configuration et tests de la communication LoRa	Juin 2024	Juillet 2024
4.3	Intégration du matériel avec le logiciel	Juillet 2024	Août 2024
4.4	Tests et validation du système IoT	Juillet 2024	Août 2024
<b>5. Documentation &amp; Finalisation</b>			
5.1	Rédaction du rapport	Août 2024	Septembre 2024
5.2	Revue et corrections	Septembre 2024	Septembre 2024
<b>6. Soutenance</b>			
6.1	Préparation de la présentation	Septembre 2024	Octobre 2024

## **ELABORATION DES MODÈLES IA**

---

## Introduction

Dans cette partie nous allons parler à propos des modèles d'intelligence artificielle utilisés pour la détection et prédiction des incendies. Nous commençons

### 1 Le modèle de détection de feu au niveau de la caméra

**TABLE 3.1 – Comparaison de MobileNet, FOMO et YOLO pour la détection d'incendie**

Caractéristique	MobileNet	FOMO	YOLO
Type	Architecture de réseau neuronal convolutif (CNN)	Modèle de détection d'objets pour microcontrôleurs	Modèle de détection d'objets
Objectif principal	Classification d'images et détection d'objets	Détection d'objets sur des appareils à faible consommation d'énergie	Détection d'objets en temps réel
Paramètres	Nombre de couches CNN, nombre de filtres par couche, fonction d'activation	Seuil de confiance, nombre max. de boîtes englobantes, IoU	Seuil de confiance, nombre max. de boîtes englobantes, IoU, paramètres d'architecture YOLO
Matériel cible	Large éventail de plateformes, y compris Raspberry Pi 4	Microcontrôleurs	Ordinateurs personnels, serveurs, appareils mobiles, Raspberry Pi 4
Avantages	Équilibre entre précision et vitesse, optimisé pour les appareils mobiles	Faible consommation d'énergie	Détection d'objets en temps réel
Inconvénients	Plus complexe que FOMO	Moins performante que MobileNet et YOLO	Moins flexible que TensorFlow

## 1.1 Les tests des modèles de Détection d’Incendie

Après avoir comparé différents modèles pour la détection d’incendie, nous avons réalisé deux essais distincts pour évaluer leur performance en conditions réelles. Le premier essai a été mené en utilisant le modèle FOMO sur la plateforme Edge Impulse. Cette plateforme nous a permis d’entraîner et de tester le modèle FOMO dans un environnement optimisé pour les microcontrôleurs et les appareils à faible consommation d’énergie. Nous avons appliqué ce modèle sur plusieurs images pour observer sa précision et sa vitesse de détection. Les résultats visuels de ces tests sont présentés ci-dessous :

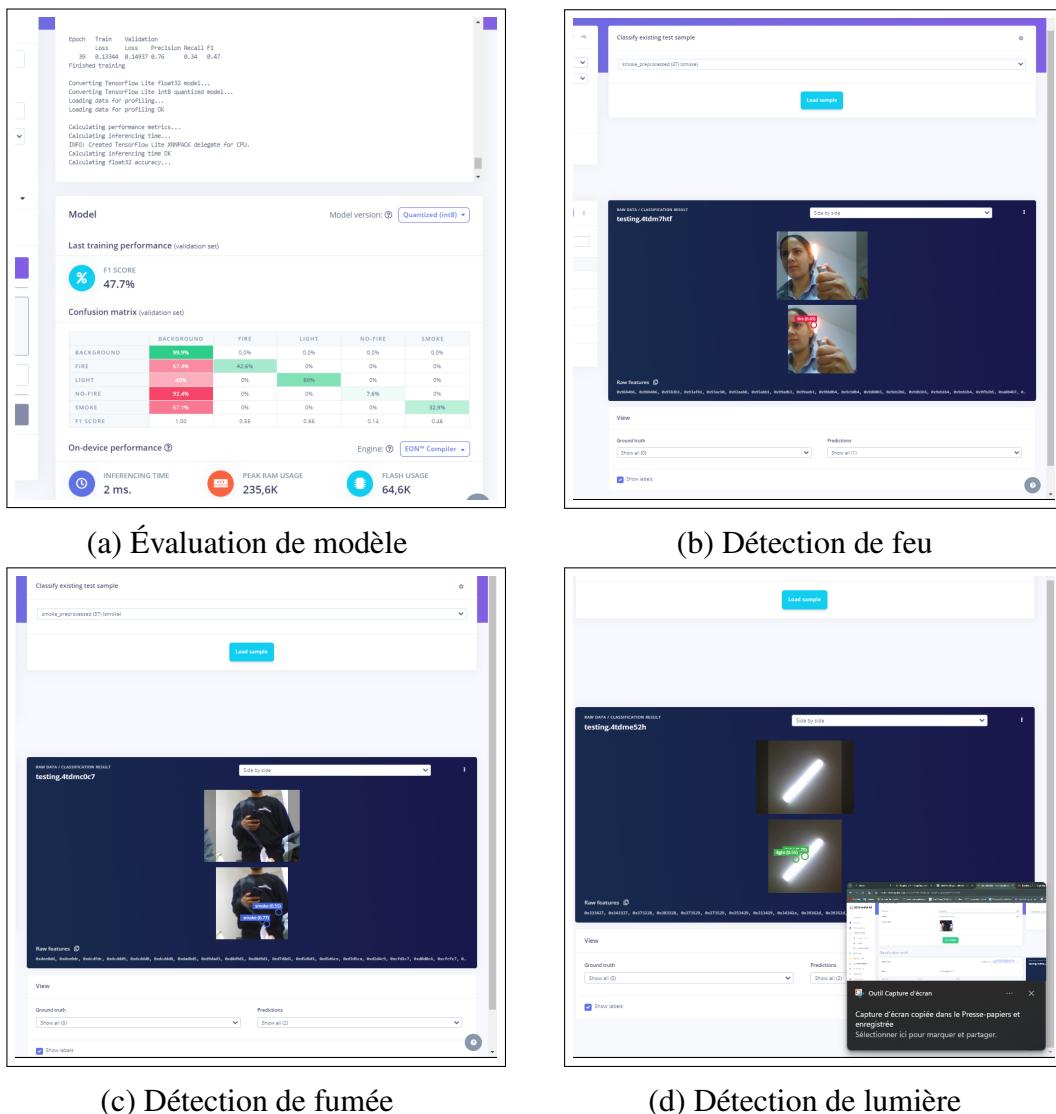


FIGURE 3.1 – Résultats des essais du modèle FOMO sur la plateforme Edge Impulse

Pour le second essai, nous avons utilisé la plateforme RoboFlow pour tester le modèle YOLO (You Only Look Once). Cette plateforme nous a permis de tirer parti des capacités avancées de YOLO pour la détection d’objets en temps réel. Nous avons appliqué le modèle sur une vidéo YouTube afin d’évaluer ses performances visuelles en direct. L’image ci-dessous

illustre les résultats obtenus lors de cet essai :

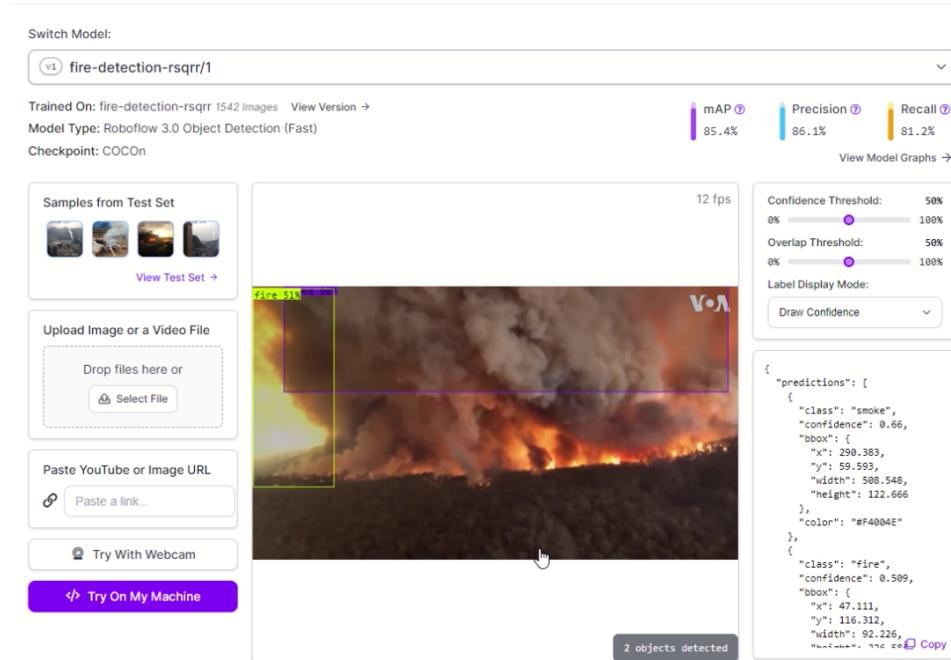


FIGURE 3.2 – Résultat visuel du modèle YOLO sur la plateforme RoboFlow

Ces essais nous ont permis de mieux comprendre les avantages et les limitations de chaque modèle dans le contexte de la détection d’incendie, et de choisir la solution la plus appropriée pour notre application.

## 1.2 Comparaison des Modèles TensorFlow Lite

Pour le choix de modèle TensorFlow Lite de MobileNet nous allons comparer les différents modèles TensorFlow lite.

TABLE 3.2 – Comparaison des Modèles TensorFlow Lite pour la Détection d’Objets

Caractéristique	SSD-MobileNet-v2	SSD-MobileNet-v2-FPNLite-320x320	EfficientDet-Lite-D0
Définition	Modèle de détection d’objets léger avec architecture CNN	Version améliorée de SSD-MobileNet-v2 avec des couches FPN pour meilleure précision	Modèle de détection d’objets optimisé pour les dispositifs Edge TPU

*Suite à la page suivante*

**TABLE 3.2 – Comparaison des Modèles TensorFlow Lite pour la Détection d’Objets (suite)**

Caractéristique	SSD-MobileNet-v2	SSD-MobileNet-v2-FPNLite-320x320	EfficientDet-Lite-D0
Paramètres	Nombre de couches CNN, fonction d’activation, quantisation INT8 disponible	Nombre de couches CNN, fonction d’activation, couches FPN, quantisation INT8 disponible	Architecture EfficientDet optimisée, quantisation INT8 disponible
Objectif Principal	Équilibre entre vitesse et précision pour des applications en temps réel	Haute précision avec compromis sur la vitesse, adapté pour des prototypes de haute précision	Performances équilibrées pour des applications sur Edge TPU et Raspberry Pi
Performances	Vitesse : 68.96 ms (quantisé) Précision : mAP 60.99% (COCO) Nombre d’objets détectés : 290/335	Vitesse : 102.67 ms (quantisé) Précision : mAP 65.24% (COCO) Nombre d’objets détectés : 306/335	Vitesse : 78.45 ms (quantisé) Précision : mAP 59.33% (COCO) Nombre d’objets détectés : 290/335
Avantages	Rapide avec une bonne précision pour des applications mobiles et embarquées	Très précis même avec des petits ensembles de données, adapté pour des applications nécessitant une haute précision	Compatible avec Edge TPU pour une accélération matérielle, bonnes performances globales
Inconvénients	Moins précis que les modèles avec FPN	Moins rapide que SSD-MobileNet-v2, nécessite plus de ressources de calcul	Moins précis que SSD-MobileNet-v2-FPNLite sur des ensembles de données limités

Les modèles TensorFlow Lite ciblent un large éventail de plateformes, offrant ainsi une flexibilité accrue pour diverses applications y compris le Raspberry Pi 4. Bien que chaque modèle présente des avantages distincts, nous avons choisi d'utiliser le modèle SSD-MobileNet-v2-FPNLite-320x320 dans notre cas particulier. Ce modèle offre une précision accrue grâce à ses couches FPN, ce qui est essentiel pour nos besoins de détection d'incendie où la précision est cruciale. De plus, il maintient une compatibilité avec des plateformes variées, y compris le Raspberry Pi 4, garantissant ainsi une flexibilité d'utilisation et une intégration aisée dans notre

système existant.

### 1.3 Étapes de création d'un modèle personnalisé de détection de feu

Dans cette section, nous détaillons le processus de création d'un modèle personnalisé pour la détection d'incendies dans des images à l'aide de TensorFlow et TensorFlow Lite. Ce travail a été réalisé dans Google Colab et comporte plusieurs étapes essentielles.

#### 1. Installation des Dépendances pour la Détection d'Objets TensorFlow

La première étape consistait à installer toutes les dépendances nécessaires, notamment TensorFlow, TensorFlow Lite, ainsi que les bibliothèques pour la détection d'objets. Ces bibliothèques incluent également des outils pour manipuler les ensembles de données et configurer l'entraînement du modèle.

#### 2. Téléchargement du Jeu de Données et Préparation des Données

Un ensemble de données d'images contenant des incendies a été téléchargé. Les images ont été redimensionnées et les étiquettes ont été associées pour être compatibles avec TensorFlow.

#### 3. Configuration de l'Entraînement

L'entraînement a été configuré en définissant les hyperparamètres tels que le nombre d'époques, la taille des lots, et le taux d'apprentissage. Un modèle de détection d'objets pré-entraîné a été utilisé comme base.

#### 4. Entraînement du Modèle Personnalisé TFLite

Le modèle a été entraîné sur le jeu de données spécifique pour apprendre à reconnaître les incendies dans les images.

#### 5. Conversion du Modèle en TensorFlow Lite

Une fois l'entraînement terminé, le modèle a été converti en TensorFlow Lite pour être déployé sur des appareils avec des ressources limitées comme le Raspberry Pi.

#### 6. Test du Modèle TensorFlow Lite

Le modèle TensorFlow Lite a été testé pour vérifier sa capacité à identifier correctement les incendies dans les images. Les résultats ont été évalués pour valider son efficacité.

#### 7. Déploiement du Modèle TensorFlow Lite

Le modèle a été déployé sur un Raspberry Pi avec une caméra, permettant la détection en temps réel des incendies pour une intervention rapide.

## 2 Le Modèle de détection d'odeurs au niveau de BME688

### 2.1 la présentation du capteur :

Le capteur BME688 est un capteur de qualité de l'air et de gaz de Bosch Sensortec, conçu pour détecter divers gaz et mesurer la température, l'humidité et la pression atmosphérique. Ce capteur multifonctionnel est particulièrement adapté pour les applications de détection d'odeurs grâce à sa haute précision, sa faible consommation d'énergie et sa capacité à détecter une large gamme de composés volatils organiques (COV). Ces caractéristiques font du BME688 un choix

idéal pour des projets de surveillance de la qualité de l'air et de détection d'odeurs dans des environnements divers.



FIGURE 3.3 – le capteur BME688

## 2.2 les étapes de développement de modèle de détection :

Le développement d'un modèle de machine learning pour détecter les odeurs à l'aide du BME688 implique plusieurs étapes clés :

### 1. Collecte des Données

Utilisation de la carte Raspberry Pi 4 pour recueillir les données du capteur BME688. Cette étape consiste à enregistrer les lectures de température, d'humidité, de pression et de gaz sur une période de temps pour capturer une variété de conditions d'odeurs et l'enregistré dans une fichier CSV.

### 2. Préparation des Données

Les données collectées sont divisées en deux ensembles distincts : un ensemble de test et un ensemble d'entraînement. Cette séparation est cruciale pour évaluer la performance du modèle de manière impartiale.

### 3. Entraînement du Modèle

Les données d'entraînement sont utilisées pour entraîner différents modèles de classification dans un environnement Jupyter Notebook. Les modèles testés incluent le **RandomForestClassifier**, le **Support Vector Classifier (SVC)** et le **KNeighborsClassifier**. Chaque modèle est évalué pour déterminer lequel offre la meilleure précision de détection des odeurs.

**RandomForestClassifier** : Un algorithme basé sur des arbres de décision, utilisant la technique de "bagging" pour créer une forêt de plusieurs arbres, chaque arbre étant

FIGURE 3.4 – Collecte des Données avec le Raspberry Pi 4

```

from sklearn.model_selection import train_test_split

# Charger les données et les séparer en X (variables) et en Y (étiquettes)
X = data[['Temperature', 'Humidity', 'Pressure', 'Gas Resistance']]
y = data['Mannual Odor Label']

# Diviser les données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=123)

print('len(X_train)')
print(len(X_train))
print(X_train.head())
print(X_train.describe())
print(X_train.info())
print('len(y_train)')
print(len(y_train))
print(y_train.head().to_string(index=False))
print('len(X_test)')
print(len(X_test))
print(X_test.head())
print(X_test.describe())
print(X_test.info())
print('len(y_test)')
print(len(y_test))
print(y_test.head().to_string(index=False))

2022
-----
Temperature Humidity Pressure Gas Resistance
2033 35.78 46.00 1014.81 3.09e+04
2034 29.12 50.98 1005.75 9.98e+03
2412 31.17 52.852 1012.68 7.393502e+05
2434 29.56 58.054 1012.00 1.179724e+06
1226 28.94 49.996 1014.82 5.477408e+05

2023
-----
smoke
gas
gas
gas
fire
B68
-----
Temperature Humidity Pressure Gas Resistance
1227 28.99 49.654 1014.81 546862.48311
1458 28.99 52.633 1014.81 621900.38697
1500 28.43 52.633 1015.83 592338.31769
776 21.31 67.828 1015.15 592335.72694
1798 28.98 58.464 1015.29 592335.72694
800
...
fire
fire
fire
smoke
Output was truncated. View as a scrollable element or open in a text editor. Adjust cell output settings.

```

FIGURE 3.5 – Préparation des Données

construit sur un échantillon aléatoire des données.

**Support Vector Classifier (SVC)** : Un algorithme qui trouve l'hyperplan optimal pour séparer les classes dans un espace de caractéristiques à haute dimension, maximisant la marge entre les classes.

**KNeighborsClassifier** : Un algorithme qui classe une donnée en fonction des classes des K données les plus proches dans l'ensemble d'entraînement.

#### 4. Évaluation des Modèles

Les performances de chaque modèle sont évaluées en utilisant l'ensemble de test. Les métriques d'évaluation incluent la précision, le rappel, et la courbe ROC pour identifier

**TABLE 3.3 – Comparaison Générale des Algorithmes**

Critère	Description
<b>Robustesse</b>	Le RandomForestClassifier tend à être plus robuste et moins sujet au surapprentissage.
<b>Complexité</b>	SVC peut être plus complexe à paramétrier et à entraîner, surtout pour les grands ensembles de données.
<b>Simplicité</b>	KNeighborsClassifier est simple à comprendre et à implémenter mais peut être inefficace pour les grands ensembles de données en termes de temps de calcul.

le modèle le plus performant.

```
... Metrics for KNeighborsClassifier model:  
Accuracy: 0.829  
Precision macro: 0.844  
Precision micro: 0.829  
Recall macro: 0.840  
Recall micro: 0.829  
F1 Score macro: 0.841  
F1 Score micro: 0.829  
Metrics for SVC model:  
Accuracy: 0.568  
Precision macro: 0.573  
Precision micro: 0.568  
Recall macro: 0.585  
Recall micro: 0.568  
F1 Score macro: 0.535  
F1 Score micro: 0.568  
Metrics for RandomForestClassifier model:  
Accuracy: 0.983  
Precision macro: 0.984  
Precision micro: 0.983  
Recall macro: 0.983  
Recall micro: 0.983  
F1 Score macro: 0.983  
F1 Score micro: 0.983
```

**FIGURE 3.6 – Évaluation des Modèles**

## 5. Enregistrement et Déploiement du Modèle

Le modèle qui donne les meilleurs résultats est enregistré sous forme de fichier .pkl (pickle file). Ce fichier est ensuite déployé sur la carte Raspberry Pi 4, permettant la détection en temps réel des odeurs grâce au capteur BME688.

```
sauvegarder le modèle  
  
import pickle  
with open('finalsaved_modelT3.pkl', 'wb') as f:  
    pickle.dump(rf_model, f)
```

**FIGURE 3.7 – Enregistrement et Déploiement du Modèle**

Ces étapes permettent de créer un système de détection d'odeurs robuste et efficace, capable de fonctionner en temps réel sur des plateformes embarquées comme le Raspberry Pi 4.

## Résumé

Le présent travail fait partie du projet de fin d'études visant à obtenir le diplôme national de master professionnel en robotique avancée et intelligence artificielle , à l'Institut Supérieur des Études Technologiques de Bizerte, pour l'année universitaire 2023/2024. L'objectif de ce projet est la conception et la réalisation d'un système IoT de détection d'incendie dans les fermes agricoles en utilisant des capteurs IoT. La communication se fera via la technologie LoRa longue portée et l'analyse des données sera réalisée en temps réel, ainsi que sur les fermes, grâce à une application web conviviale.

Mots clés : UML, SysML, Scrum, Raspberry, LoRa, Django, PostgreSQL, GeoDjango, ML.

## Abstract

This work is part of the final year project aimed at obtaining the National Diploma of Professional Master degree in advanced robotics and artificial intelligence , at the Higher Institute of Technological Studies of Bizerte, for the academic year 2023/2024. The objective of this project is the design and implementation of an IoT system for fire detection in agricultural farms using IoT sensors. Communication will be done via long-range LoRa technology, and data analysis will be performed in real time, as well as on farms, thanks to a user-friendly web application.

Keywords : UML, SysML, Scrum, Raspberry, LoRa, Django, PostgreSQL, GeoDjango, ML.