



# **DeepRice**

## **Agriculture numérique: Utilisation de l'IoT et de l'IA dans la riziculture**

Par

MONG SANAHARISOA Adrian Jeffrey

Mémoire présenté

MASTER of SCIENCE eBIHAR de l'ESTIA

Novembre, 2023

Jury :

M. Hu Fai Steave LEONG, encadreur professionnel

M. Koto RAKOTODRALAMBO, encadreur pédagogique

© MONG SANAHARISOA Adrian Jeffrey, 2023



## Résumé

De nos jours, l'agriculture numérique émerge comme un secteur en croissance, exploitant les avancées technologiques pour améliorer l'efficacité et la productivité agricole. Notre objectif consiste à instaurer une plateforme au sein d'une exploitation agricole pilote. Cette plateforme aura pour fonctionnalités le suivi et la gestion d'une parcelle de rizière, ainsi qu'une assistance destinée aux agronomes pour résoudre d'éventuelles anomalies. Le présent document se penchera sur une analyse de l'existant et proposera la mise en place d'un système IoT. Nous élaborerons une architecture de Data Lake à la fois open source et économique à destination des agriculteurs. Nous entreprendrons également une comparaison pour sélectionner une méthode basée sur les réseaux neuronaux convolutifs (CNN) en vue de détecter les anomalies sur les feuilles de riz. Parallèlement, nous recourrons à l'apprentissage profond (Deep Learning) pour anticiper les besoins en irrigation en utilisant les données collectées par les capteurs ainsi que les informations météorologiques.

## Abstract

Nowadays, digital agriculture is emerging as a growing sector, harnessing advances in technology to improve agricultural efficiency and productivity. Our goal is to establish a platform within a pilot farm. This platform will have the functions of monitoring and managing a plot of rice field, as well as assistance for agronomists to resolve any anomalies. This document will focus on an analysis of the existing situation and will propose the implementation of an IoT system. We will build an open-source, cost-effective Data Lake architecture for farmers. We will also undertake a comparison to select a method based on convolutional neural networks (CNN) to detect anomalies on rice sheets. At the same time, we will use deep learning to anticipate irrigation needs using data collected by sensors and weather information.

# Table des matières

Liste des tableaux .....	iv
Liste des figures.....	v
Glossaire .....	vii
Avant-propos .....	9
L'IT University.....	9
Bureau International de Consultance en Informatique .....	9
Remerciements .....	10
Introduction .....	11
1    Présentation du stage .....	12
1.1    Présentation du sujet et objectifs du projet (problème traité).....	12
2    État de l'art sur le sujet traité.....	12
2.1    Critères de comparaison .....	13
2.1.1    Couche de perception .....	13
2.1.2    Couche réseau.....	14
2.1.3    Couche de traitement de données .....	15
2.1.4    Couche applicative .....	16
2.1.5    Couche de sécurité.....	16
2.2    Étude des projets sur l'agriculture numérique .....	16
2.3    Conclusion.....	18
3    Étude de l'existant et solution envisagée.....	18
3.1    Étude de l'existant .....	18
3.2    Solutions envisagées.....	19
3.3    Objectifs principaux et livrables.....	20
4    Démarche projet .....	20
4.1    Principes de la démarche projet.....	20

4.1.1	Activités d'ingénierie logicielle.....	20
4.1.2	Méthode de gestion de projet utilisée .....	21
4.1.3	Rôles et responsabilités .....	21
4.1.4	Outils .....	21
4.1.5	Gestion de la configuration.....	22
4.2	Contraintes et risques sur le projet .....	23
4.3	Démarche projet mise en œuvre .....	23
4.4	Planification.....	24
4.5	Budget du projet .....	26
5	Exigences réalisées dans le projet (vision externe/utilisateur) .....	26
5.1	Exigences fonctionnelles – Cas d'utilisation.....	26
5.1.1	Détection d'anomalies du riz à partir d'images et suggestion de mesure à prendre .....	26
5.1.2	Monitoring agricole .....	28
5.1.3	Anticipation d'irrigation .....	29
5.2	Exigences non fonctionnelles transverses .....	31
5.2.1	Mise en place d'un Data Lake .....	31
5.2.2	Performance.....	32
5.2.3	Maintenabilité.....	32
5.3	Interfaces détaillées .....	33
5.3.1	IHM .....	33
5.3.2	Interfaces avec d'autres systèmes.....	36
6	Architecture(s) système .....	38
7	Conception du système logiciel réalisée dans le projet (vision interne/développeur). 39	
7.1	Plate-forme technique.....	39
7.2	Conception du logiciel développé .....	41

7.2.1	Conception du code source.....	41
7.2.2	Conception du modèle de classification d'anomalies Deep Learning.....	42
7.2.3	Le code source – vue statique.....	47
7.2.4	Les composants et leur déploiement.....	47
8	Tests du système logiciel.....	48
9	Conclusion générale .....	48
10	Références et Bibliographie.....	50
11	Annexes .....	52
11.1	Annexe 1 : Interface interaction du Data Lake dans la machine virtuelle.....	52
11.2	Annexe 2 : Comparaison des sources de données .....	53
	Annexe 3 : Courbe d'apprentissage du modèle .....	57
11.3	Annexe 4 : Comparaison de R et Python.....	58

## Liste des tableaux

Tableau 1. Projets existants sur l'agriculture numérique .....	17
Tableau 2. Outils utilisés .....	22
Tableau 3. Contraintes et risques sur le projet.....	23
Tableau 4. Budget du projet .....	26
Tableau 5. Conception du Data Lake .....	32
Tableau 6. Comparaison entre Google drive et Wasabi .....	54
Tableau 7. Tableau comparatif des sources de données .....	56
Tableau 8. Comparaison entre R et Python .....	58

## Liste des figures

Figure 1. Planning de réalisation .....	25
Figure 2. Diagramme de séquence - Détection d'anomalie .....	28
Figure 3. Diagramme de séquence - Monitoring agricole .....	29
Figure 4. Diagramme de séquence - Anticipation d'irrigation .....	31
Figure 5. Écran outils .....	33
Figure 6. Écran de détection d'anomalies .....	33
Figure 7. Résultat de la détection d'anomalie .....	34
Figure 8. Discussion avec l'agriculteur virtuel .....	34
Figure 9. Ecran d'accueil/Monitoring .....	35
Figure 10. Ecran des notifications .....	36
Figure 11. Paramètres de l'application.....	36
Figure 12. Open-Meteo - Configuration des données météorologique à récupérer .....	37
Figure 13. Open-Meteo - Visualisation des données.....	37
Figure 14. Architecture du système.....	39
Figure 15. Feuille saine .....	43
Figure 16. Tache brune.....	43
Figure 17. Rouille des feuilles.....	43
Figure 18. Blast du riz .....	43
Figure 19. Modèle séquentiel .....	44
Figure 20. Evaluation: Modèle basé sur le modèle séquentiel .....	44
Figure 21. Modèle basé sur le modèle Xception .....	45
Figure 22. Evaluation: Modèle basé sur le modèle Xception.....	45
Figure 23. Modèle basé sur le modèle MobileNetV2.....	46
Figure 24. Structure code source: Machine learning .....	47
Figure 25. Structure code source: Backend .....	47



Figure 26. Structure code source: Frontend (application mobile) .....	47
Figure 27. Liste de machine virtuelle dans virtualBox.....	52
Figure 28. Connexion à clickhouse .....	52
Figure 29: Connexion avec Cassandra .....	53
Figure 30. Démarrage de HDFS .....	53
Figure 32. Comparaison prix sur les stockages cloud .....	55
Figure 34. Courbe d'évolution de la précision du modèle .....	57
Figure 35. Courbe d'évolution du taux de perte du modèle.....	57

# Glossaire

**IoT** : Ou Internet des objets, c'est un réseau d'objets et de terminaux équipés de capteurs qui permet de recevoir et de transmettre des données avec d'autres systèmes.

**IA** : Intelligence artificielle, processus d'imitation de l'intelligence humaine qui repose sur la création et l'application d'algorithmes exécutés dans un environnement informatique dynamique. Son but est de permettre à des ordinateurs de penser et d'agir comme des êtres humains.

**Data Lake** : système de stockage de données qui permet de stocker une grande quantité de données brutes, structurées ou non structurées, dans leur format d'origine

**API** : ou Application Programming Interface, c'est un ensemble de règles et de protocoles qui permettent à différentes applications informatiques de communiquer entre elles.

**Dataset** : collection de données, généralement présentée sous forme de table, de fichier ou d'ensemble de documents, qui est utilisée à des fins d'analyse, de recherche, de traitement ou d'étude.

**SOLID** : ensemble de cinq principes de conception logicielle qui visent à créer des logiciels plus robustes, flexibles et faciles à maintenir.

**Blockchain** : est une technologie qui permet de garder la trace d'un ensemble de transactions, de manière décentralisée, sécurisée et transparente, sous forme d'une chaîne de blocs.

**Test unitaire** : une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme.

**Test d'intégration** : un test qui se déroule dans une phase d'un projet informatique suivant les tests unitaires. Il consiste, une fois que les développeurs ont chacun validé leurs développements ou leurs correctifs, à regrouper leurs modifications ensemble dans le cadre d'une livraison.

**Méthodologie en cascade** : Méthodologie de gestion de projet séquentielle qui divise les projets en plusieurs phases distinctes.

**IDE** : un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels.



## **Avant-propos**

Le présent mémoire présente les résultats du travail effectué lors de mon stage de fin d'études de Master eBIHAR en Informatique de l'IT University, stage effectué à BICI durant 3 mois, de juillet à septembre 2023.

Afin de poser clairement le contexte de ce mémoire, je vais présenter succincte d'une part de l'IT University et d'autre part de mon institution/entreprise d'accueil.

## **L'IT University**

Fondée en 2011, l'IT University (ou ITU) est une université privée, spécialisée en informatique, formant les jeunes bacheliers scientifiques :

- En trois ans, pour l'obtention d'une Licence, option Développement d'applications, Réseaux et Bases de Données ou Web et Design
- En cinq ans, pour l'obtention d'un Master MBDS en coopération avec l'Université de Nice Sophia Antipolis - France

Étant une formation professionnalisante, l'ITU a tissé des liens forts avec ses partenaires industriels, dont l'opérateur convergent TELMA, le GOTICOM et la plupart des entreprises et institutions du secteur des TIC<sup>1</sup>. Ces partenaires participent effectivement à la formation par la fourniture de connexion Internet à haut débit, l'envoi de conférenciers ou encore par l'accueil des étudiants en stage. D'autre part, le corps enseignant de l'ITU est constitué intégralement de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs. Enfin, l'ITU fait partie du programme Microsoft Imagine en plus d'être un Oracle Gold Partner.

## **Bureau International de Consultance en Informatique**

Le BICI ou Bureau International de Consultance en Informatique est une branche du groupe Vidy Varotra, une société de Service en Ingénierie Informatique Malgache créée en

---

<sup>1</sup> TIC : Technologies de l'Information et de la Communication

2003, sous l'appellation JRews Informatique à l'époque. Sa principale activité est l'assistance informatique et BICI offre aujourd'hui les services suivants :

- Ingénieur logiciel : développement spécifique, mobile et site web
- Ingénierie de performance et sécurité
- Réseau, communication et système d'exploitation
- Solution matérielle : dépannage, location et restauration de matériel
- Audit Informatique
- Formation (en partenariat avec l'IT University)

## **Remerciements**

Premièrement, je tiens à remercier toute l'équipe pédagogique de l'ESTIA et de l'IT-University, et particulièrement Monsieur Serge MIRANDA qui est le fondateur du Master of Science BIHAR.

Je remercie également BICI de m'avoir accepté et formé durant ces 3 mois de stage. Un remerciement distingué pour mon encadreur professionnel Monsieur Hu Fai Steave LEONG, pour ses conseils et le partage de son expertise au quotidien. Grâce à tout cela, j'ai pu accomplir pleinement mes missions en évitant certaines difficultés.

Je tiens aussi à remercier sincèrement les membres du jury qui me font le grand honneur d'évaluer ce travail.

Je n'oublie pas d'adresser mes remerciements à ma famille, mes amis et mes collègues pour le soutien, les encouragements, la patience et la compréhension.

Pour finir, je tiens à remercier toutes les personnes qui m'ont fait part de leurs judicieux conseils lors de la réalisation de ce rapport de stage.

# Introduction

Selon la Banque mondiale, le rendement rizicole à Madagascar est de 2,45 tonnes par hectare ce qui est largement inférieur à la moyenne mondiale qui est de 4,8 tonnes par hectare [21]. Avec l'aide de l'IoT et de l'IA qui font déjà parti de notre quotidien dans plusieurs et différents domaines, nous pourrions construire une agriculture numérique qui révolutionne nos champs et nos récoltes. Mais comment faire usage de l'IA et de l'IoT dans vos champs ?

« Des chiffres aux récoltes, une transformation digitale fertile » - notre slogan qui annonce une aventure au cœur de l'efflorescence agricole moderne.

DeepRice est une solution numérique destinée aux agriculteurs qui fera usage de l'IA et de l'IoT afin d'améliorer le rendement agricole et d'optimiser l'utilisation des ressources. Pour cela, il sera possible de détecter les anomalies et maladies du riz à partir d'images, anticiper les besoins d'irrigation grâce aux données recueillies par les capteurs et les données météorologiques afin d'optimiser l'utilisation des ressources qui sont limitées. Ajouter à cela, les agriculteurs auront un agriculteur virtuel à leur disposition qui sera alimenté par ChatGPT-4 afin de les aider à réagir face aux difficultés.

Pour entrer dans le vif du sujet, ce rapport de stage sera divisé en plusieurs parties distinctes :

- En premier lieu, nous commencerons par une présentation du sujet, du stage et des objectifs du projet.
- Nous ferons ensuite l'état de l'art et une analyse de l'existant sur le sujet qui sera suivi de la démarche choisie pour la réalisation du projet.
- Dans les chapitres suivants, nous verrons la conception, l'architecture ainsi que les réalisations faites durant ce stage. Et enfin, nous finirons par une conclusion.

# **1 Présentation du stage**

## **1.1 Présentation du sujet et objectifs du projet (problème traité)**

L'agriculture numérique est l'utilisation des technologies de l'information et de la communication (TIC) pour améliorer la productivité, la durabilité et la résilience de l'agriculture. L'IoT et l'IA sont deux des TIC les plus importantes utilisées dans l'agriculture numérique.

L'IoT désigne le réseau d'objets physiques dotés de capteurs, de logiciels et d'une connectivité réseau pour collecter et échanger des données. L'IA désigne la capacité des machines à apprendre et à prendre des décisions sans être explicitement programmées.

Sur le plan métier, cette plateforme se positionne comme un outil stratégique pour les exploitations agricoles, leur permettant de franchir un cap technologique significatif. Elle se présente comme une solution prometteuse pour faire face aux défis posés par les aléas climatiques et les exigences d'une agriculture de précision.

Les principaux enjeux du projet se cristallisent autour de l'amélioration de la productivité agricole, de la réduction des coûts d'exploitation, et de la gestion durable des ressources, notamment de l'eau. La mise en œuvre d'une telle plateforme n'est cependant pas exempte de risques. Il s'agit notamment de garantir l'acceptation et l'adoption de ces technologies par les agriculteurs, d'assurer la sécurité et la confidentialité des données agricoles, et de maintenir une connectivité réseau stable pour le bon fonctionnement des capteurs et le traitement des données.

En définitive, le projet proposé dans ce mémoire s'avance comme une initiative pionnière qui pourrait considérablement transformer l'agriculture à Madagascar, en la rendant plus intelligente, plus réactive aux besoins des cultures, et plus respectueuse de l'environnement.

## **2 État de l'art sur le sujet traité**

Dans le cadre de la compréhension approfondie des avancées actuelles et de la place du projet dans le spectre de l'agriculture numérique, il est impératif d'examiner l'état

de l'art. Ce chapitre se consacre à l'analyse détaillée des technologies, des méthodologies et des pratiques prévalentes dans le domaine de l'agriculture intelligente. Nous passerons en revue les systèmes et les outils existants, les innovations récentes dans le domaine de l'IoT appliqué à l'agriculture, en mettant l'accent sur leur impact sur la productivité et la durabilité agricoles.

Cet examen critique nous permettra non seulement de situer notre projet dans le contexte de la recherche actuelle mais aussi d'identifier les lacunes et les opportunités pour innover et apporter une contribution significative à ce secteur en pleine transformation. Ainsi, la réalisation d'une telle application nécessite l'utilisation d'un modèle d'architecture IoT qui généralement est composée des couches suivantes :

- Couche de perception
- Couche de réseau
- Couche de traitement de données
- Couche applicative
- Couche de sécurité

## **2.1 Critères de comparaison**

### **2.1.1 Couche de perception**

Un capteur est un dispositif qui convertit une grandeur physique, telle que la température, l'humidité ou la luminosité, en un signal électrique ou optique. Ce signal peut ensuite être transmis à un système de contrôle ou d'acquisition de données.

Les capteurs sont utilisés dans de nombreux domaines, tels que l'industrie, la médecine, l'agriculture, la domotique, etc. Ils permettent de collecter des données sur l'environnement, les machines, les personnes ou les animaux. Ces données peuvent ensuite être utilisées pour améliorer les performances, la sécurité ou le confort.

- Capteurs du sol : Ils mesurent des paramètres tels que l'humidité du sol, la température, la salinité, le pH, la conductivité électrique, etc. Ces données aident à optimiser l'irrigation, la fertilisation et la gestion globale du sol.
- Capteurs météorologiques : ils mesurent des paramètres climatiques tels que la température de l'air, l'humidité relative, la vitesse et la direction du vent, la



pluviométrie, etc. Ces données sont essentielles pour comprendre les conditions météorologiques locales et leur impact sur les cultures

- Capteurs de cultures : Ils permettent de surveiller les niveaux de nutriments dans les plantes (l'azote, le phosphore, le potassium, etc.), la croissance, la santé, la maturité et bien d'autres choses. Ces capteurs peuvent aider à optimiser l'apport en nutriments, à éviter la sur ou sous fertilisation et mesurer l'état des cultures.
- Capteurs de qualité de l'eau : Ils permettent de mesurer la qualité de l'eau pour l'irrigation et d'autres utilisations agricoles. Ils peuvent détecter la présence de contaminants, de substances chimiques indésirables.
- Capteurs d'imagerie : Ces capteurs capturent des images à plusieurs longueurs d'onde, ce qui permet d'analyser la santé des cultures, la couverture végétale, la détection du stress, etc. Ils sont souvent utilisés dans la télédétection et les drones agricoles.

### **2.1.2 Couche réseau**

Les échanges d'informations parmi les capteurs, les dispositifs et les plateformes cloud peuvent impliquer une variété de technologies distinctes. Dans cette optique, nous présenterons de manière concise les systèmes de communication fréquemment adoptés dans l'univers de l'Internet des Objets (IoT). Il est pertinent de mentionner que la sélection d'une méthode de communication est fortement influencée par les ressources accessibles à chaque localité.

- Réseaux cellulaires : Les capteurs peuvent se connecter aux réseaux de communication mobiles tels que le GSM, la 3G, et 4G LTE, voire le 5G dans certains cas. Ils utilisent des cartes SIM pour établir une connexion avec les tours cellulaires et transmettre les données à distance.
- Réseaux sans fil à courte portée : Certains capteurs utilisent des technologies sans fil à courte portée, telles que le Wi-Fi (802.11) ou le Bluetooth, pour communiquer avec une station de base ou une passerelle à proximité. Ces technologies sont souvent utilisées pour les capteurs à faible consommation d'énergie ou pour les déploiements à petite échelle.
- Réseaux LPWAN (Low-Power Wide Area Network) : Les réseaux LPWAN, tels que LoRaWAN et Sigfox, sont conçus spécifiquement pour les applications IoT à faible consommation d'énergie et à longue portée. Ils permettent aux capteurs

d'envoyer des données sur de longues distances avec une consommation d'énergie réduite, ce qui est avantageux dans les zones rurales ou les grandes exploitations agricoles.

- Réseaux satellites : Dans les zones où la connectivité cellulaire est limitée, les capteurs peuvent utiliser des réseaux de communication par satellite pour transmettre les données. Les communications par satellite offrent une couverture mondiale, mais peuvent être plus coûteuses et nécessiter des équipements spécifiques.
- Réseau LoRaWan : LoRa est une technologie de modulation RF pour les réseaux étendus à faible puissance (LPWAN). LoRaWAN (Long Range Wide Area Network) est un protocole de réseau étendu à faible consommation construit sur la base de la technique de modulation radio LoRa. Il connecte sans fil les appareils à internet et gère la communication entre les appareils des nœuds finaux et les passerelles de réseau. LoRaWAN est conçu pour être utilisé dans les applications de l'internet des objets (IoT), où une longue durée de vie de la batterie et une communication à longue portée sont essentielles.

### **2.1.3 Couche de traitement de données**

La couche de traitement des données, un composant crucial dans l'architecture des systèmes IoT, peut être orchestrée via deux approches distinctes :

- Premièrement, les solutions cloud, fournies par des géants technologiques tels que Google, Microsoft, et Amazon, sont fréquemment choisies pour leur efficacité dans la gestion des sollicitations émanant des appareils IoT. Ces plateformes assurent la sécurité, une communication sûre et la disponibilité des données.
- En deuxième lieu, l'implémentation de la logique de traitement des données peut être réalisée au moyen de serveurs dédiés ou de machines virtuelles, permettant une personnalisation accrue. Comme nous travaillerons avec divers types de données avec un volume qui augmentent avec le temps, il est essentiel d'avoir une méthode de stockage et de traitement efficace pour divers formats de données. Il est alors nécessaire de mettre en place un Data Lake permettant la gestion et la manipulation aisée de ces données diversifiées et simplifiant les processus de traitement subséquents.

### **2.1.4 Couche applicative**

La couche applicative constitue l'interface de l'utilisateur, offrant des services spécifiques à son application. Elle se déploie à travers des applications mobiles ou web, permettant d'accéder aux données de l'appareil et d'afficher les résultats de l'analyse. Les applications IoT ont souvent recours à des API pour s'intégrer à la couche de traitement des données, tandis que certaines plates-formes IoT offrent une infrastructure logicielle prête à l'emploi permettant l'exploration des données, l'analyse avancée et la visualisation.

### **2.1.5 Couche de sécurité**

La sécurité occupe une place fondamentale dans une application d'IoT dédiée à l'agriculture numérique. Dans ce domaine en pleine expansion, où les dispositifs connectés et l'IoT sont largement utilisés, il est crucial de protéger les données et de prévenir les attaques cybernétiques. La couche de sécurité traverse toutes les couches précédentes.

La sécurité de l'IoT agricole constitue une préoccupation de premier plan, car les logiciels malveillants et les vols de données peuvent perturber gravement le bon fonctionnement des systèmes connectés. Avec l'augmentation du nombre d'appareils IoT utilisés dans l'agriculture, les risques d'attaques malveillantes se multiplient également.

Il est donc impératif d'établir des politiques de sécurité spécifiques à l'agriculture numérique. Une approche prometteuse pour renforcer la sécurité consiste à tirer parti de la technologie de la blockchain et des contrats intelligents.

En utilisant les contrats intelligents de la blockchain, il est possible d'améliorer la sécurité du protocole de routage en réduisant la redondance des données agrégées collectées par les nœuds IoT. Cette approche permet de contrer les attaques potentielles sur le réseau IoT tout en optimisant la consommation d'énergie, prolongeant ainsi la durée de vie du réseau.

## **2.2 Étude des projets sur l'agriculture numérique**

De nombreuses initiatives innovantes ont vu le jour dans le monde entier, chacune apportant sa propre perspective et sa méthodologie pour révolutionner l'agriculture. Nous entreprendrons une analyse de ces projets sélectionnés, en scrutant leurs caractéristiques, ainsi que leurs avantages et limites.

Caractéristiques	Phytech [20]	FASAL [19]	Deep Farming [1]
<b>Pays</b>	Israël	Inde	Côte d'Ivoire, Madagascar, Turquie, Bénin, République Dominicaine, Haïti
<b>Application</b>	Optimisation de l'irrigation	Prévision des récoltes	Améliorer le rendement, réduire la consommation d'eau
<b>Entreprise</b>	Phytech	ICAR-Central Research Institute for Dryland Agriculture	DATA LAB de l'ESTIA et des CAN BIHAR
<b>Technologie</b>	Capteurs, analyse de données	Intelligence artificielle, analyse de données	Intelligence artificielle, analyse de données
<b>Capteurs utilisés</b>	Capteurs IoT sur les plantes pour mesurer la tension de l'eau	Imagerie satellitaire, données météorologiques	Capteurs IoT de SINAFIS, drones, satellites
<b>Avantages</b>	Gestion précise de l'irrigation, réduction du stress hydrique des plantes, optimisation des rendements	Augmentation des rendements, réduction de l'utilisation de l'eau et des pesticides, amélioration de la qualité	Efficacité accrue de l'agriculture, réduction de l'impact sur l'environnement, prévention des maladies et des ravageurs, accès abordable pour les petits agriculteurs
<b>Faiblesses</b>	Dépendance à la connectivité Internet, possibilité de facteurs non liés à l'eau influençant les recommandations d'irrigation	Précision des données, couverture limitée, manque de transparence, coût élevé	Dépendance aux technologies, besoin de compétences techniques, limitations des modèles d'apprentissage automatique

Tableau 1.Projets existants sur l'agriculture numérique

## **2.3 Conclusion**

L'examen de l'état actuel de l'agriculture numérique a révélé un certain nombre d'options, mais aussi des défis. Les défis identifiés sont les suivants :

- La connectivité dans les zones rurales : un système d'agriculture numérique nécessite une connexion Internet rapide et fiable. Or, la connectivité Internet est souvent indisponible dans les zones rurales.
- Le manque de données : Afin de mettre en œuvre notre application, nous avons besoin d'un ensemble de données afin de concevoir les modèles d'apprentissage automatique or que Madagascar ne possède aucune données exploitables sur la riziculture.
- La compatibilité entre différentes technologies : les différentes technologies utilisées dans l'agriculture numérique doivent être compatibles entre elles, que ce soit au niveau des capteurs ou de la plateforme IoT.

Le problème de connectivité est particulièrement important dans les zones rurales. En effet, les zones rurales sont souvent plus éloignées des centres urbains, où la connectivité Internet est plus développée. Cela peut rendre difficile la mise en œuvre d'un système d'agriculture numérique dans ces zones. Cependant, il existe des solutions potentielles à ce problème. Par exemple, le réseau satellitaire StarLink [22] de SpaceX pourrait fournir une connectivité Internet rapide et fiable partout dans le monde.

Pour résoudre la compatibilité entre différentes technologies, on peut utiliser un système IoT tout prêt. Ce système peut fournir les capteurs, la plateforme ou les deux. Cependant, notre objectif est de proposer une agriculture numérique open source et low cost. Par conséquent, il est préférable de concevoir notre propre système IoT.

## **3 Étude de l'existant et solution envisagée**

### **3.1 Étude de l'existant**

La culture du riz est cruciale pour l'économie de Madagascar, représentant environ 30 % de la production agricole et constituant 80% de la consommation de céréales du pays, tout en fournissant de l'emploi à près de 5 millions de personnes. Néanmoins, malgré l'implication massive de la population, la production reste inférieure à la moyenne

mondiale, avec seulement 2,45 tonnes par hectare contre 4,8 tonnes internationalement [21].

Face à ce défi, le pays a vu ses importations de riz bondir de 35 % entre 2022 et le début de 2023, passant de 113 299 à 151 322 tonnes [3], reflétant les lacunes de la production locale. Cette situation est exacerbée par les méthodes de culture variées mais souvent peu productives, allant des systèmes inondés traditionnels aux pratiques pluviales et biologiques, chacune confrontées à des obstacles tels que des aléas climatiques, des ressources limitées et une vulnérabilité aux maladies et parasites. Néanmoins, les techniques comme la riziculture biologique sont respectueuses de l'environnement car elles n'utilisent pas d'engrais chimique et de pesticide.

Les vulnérabilités vues auparavant sont aggravées par le manque de contrôle et d'innovation, qui pourrait autrement limiter la diffusion de maladies réduisant encore plus les rendements. L'utilisation intensive d'engrais chimiques et de pesticides aggrave la situation en polluant l'eau, le sol et l'air. De surcroît, une gestion inadéquate des ressources essentielles comme l'eau et les fertilisants se traduit par une hausse des coûts de production, érodant davantage la marge bénéficiaire des riziculteurs.

### **3.2 Solutions envisagées**

Confrontée aux défis de l'agriculture malgache, l'adoption de technologies agricoles numériques est essentielle. Nous suggérons les solutions suivantes :

- Procéder à une analyse de données approfondie pour cerner les problèmes existants et recueillir les retours des agriculteurs, ce qui permettra de comprendre leurs besoins et les difficultés auxquelles ils sont confrontés.
- Assister les agriculteurs dans la détection précoce des anomalies ou maladies affectant les cultures de riz, et fournir immédiatement des recommandations pour les actions correctives à prendre afin de prévenir les pertes de récoltes.
- Accompagner les riziculteurs dans une meilleure gestion de leurs ressources. La gestion efficace de l'eau et des engrais est cruciale en agriculture, et face à l'augmentation globale de la rareté de l'eau, il est vital de minimiser le gaspillage et de planifier l'irrigation aux moments les plus opportuns.

### 3.3 Objectifs principaux et livrables

En conséquence, les buts principaux de l'application, issus des solutions suggérées, seront :

- Le développement d'une infrastructure de Data Lake open source et low cost pour le stockage des données qui seront recueillis via les différents capteurs.
- L'établissement d'un système de monitoring pour surveiller l'état actuel des champs en direct et prévoir les besoins d'irrigation futurs.
- La création d'un outil permettant de détecter les anomalies dans les cultures de riz via analyse d'image et de recommander des actions correctives appropriées.

## 4 Démarche projet

### 4.1 Principes de la démarche projet

#### 4.1.1 Activités d'ingénierie logicielle

Les phases d'ingénierie logicielle à mettre en œuvre pour le projet incluent : la définition des exigences, la conception, le développement, les tests et la validation, ainsi que la maintenance. Voici un aperçu de chaque phase :

- Définition des exigences : cette phase vise à déterminer les besoins spécifiques du projet par rapport à l'existant et à rédiger un cahier des charges pour guider les travaux à venir.
- Conception : cette phase implique de planifier et de schématiser les composants nécessaires pour répondre au cahier des charges.
- Développement : cette étape concerne la construction effective du logiciel, en codant les fonctionnalités requises en conformité avec le plan de conception.
- Tests et validation : après chaque phase de développement, des tests unitaires et tests d'intégrations sont effectués par les développeurs, suivis de tests de validation en collaboration avec le chef de projet.

#### 4.1.2 Méthode de gestion de projet utilisée

Pour ce projet, nous avons choisi de suivre la méthodologie en cascade, qui implique de finaliser chaque phase avant de passer à la suivante. En tant que membre de l'équipe de développement, j'ai activement participé à des réunions hebdomadaires pour définir les exigences et surveiller la progression du projet.

#### 4.1.3 Rôles et responsabilités

Voici les diverses parties impliquées dans le projet :

- Le client : responsable de la définition des besoins, de la validation des livrables, des retours continus et de la collaboration avec l'équipe du projet.
- Un enseignant de l'université d'agronomie d'Antananarivo : chargé de fournir des connaissances en agriculture et de faciliter la liaison avec d'autres parties prenantes du projet.
- Un ancien étudiant du Master eBIHAR : apporte son expertise dans le domaine de l'agriculture numérique.
- Les contributeurs de données : responsable de la fourniture de données pour l'application ou un domaine spécifique.

Enfin, l'équipe du projet se compose de :

- Le chef de projet : responsable de la gestion de l'avancement du projet et de la mise à disposition des ressources nécessaires à l'équipe.
- Deux développeurs : dont moi-même et un autre stagiaire en master MBDS, chargés de la conception et de la réalisation du projet.

#### 4.1.4 Outils

Voici les outils qui ont été utilisés pour réaliser ce projet :

Outil	Utilisation	Description
Google Meet	Pour les réunions avec le client	Outil puissant et facile à utiliser qui peut être utilisé pour des réunions, des présentations, des formations et bien plus encore



Skype	Pour communiquer avec l'équipe	Service de communication en ligne qui permet de passer des appels audio et vidéo, d'envoyer des messages instantanés, de partager des fichiers et de participer à des conversations en groupe
Gitlab	Contrôle de versions	Plateforme de gestion de code source et de collaboration en équipes
Draw.io	Création de diagramme	Application de création de divers diagrammes en ligne avec une interface intuitive pour l'utilisateur
Jupyter notebook	Développement des modèles d'apprentissage automatique	Application dans le domaine de la science des données apprentissage automatique. Possède une interface interactive qui facilite grandement l'utilisation
Visual Studio Code	Développement de l'application serveur	IDE open source hautement personnalisable développé par Microsoft. Conçu pour coder sous différents langages
Android Studio	Développement de l'application mobile	IDE officielle des applications Android. Basé sur le puissant outil de développement et d'édition de code d'IntelliJ IDEA
Postman	Pour tester l'API	Application de test pour les API, facile d'utilisation qui permet de personnaliser les requêtes

Tableau 2. Outils utilisés

#### 4.1.5 Gestion de la configuration

Les documents du projet sont stockés dans un dossier partagé sur Google Drive et pour contrôler les versions du code, l'équipe dispose de trois dépôts GitLab, à savoir :

- Machine Learning : dédié au code pour la création des modèles d'apprentissage automatique.
- Backend : réservé au code source de l'application serveur.
- Frontend : dédié à l'application mobile.

Voici l'organisation des branches pour chacun de ces dépôts :

- Les branches **feature/nom-fonctionnalité** : où les nouvelles fonctionnalités sont développées.
- Les branches **bugfix/description** : qui contient les correctifs des bugs remontés.
- La branche **dev** : où sera mergé les implémentations faites par les développeurs dans leur branche respective et où sera fait les tests ainsi que les correctifs, le travail effectué sur cette branche doit être testé et validé par les développeurs et le chef de projet.
- La branche **main** : la branche principale du projet où sera mergé le travail qui a été validé par l'équipe du projet.

## 4.2 Contraintes et risques sur le projet

Libellé du risque	Importance	Facteur contribuant	Solutions proposées
Capteurs indisponibles	Important	Pas de marchand ou d'entreprise fournisseur/constructeur des capteurs requis	Importer des capteurs de l'étranger
Incapacité d'étudier les données locales et de collecter des données durant la période du stage	Important	Aucune donnée disponible	Utiliser les données du web si possible. Pour l'anticipation d'irrigation, utiliser l'équation de Penman-Monteith [23] si aucune données
Connectivité limitée dans les zones rurale	Important	La couverture réseau dans certaines régions est limitée et peut conduire à une connexion instable ou voire inexistante	Utilisation du réseau satellitaire Starlink de SpaceX lors de sa disponibilité

Tableau 3. Contraintes et risques sur le projet

## 4.3 Démarche projet mise en œuvre

### Phase 1 : Définition des Exigences

Dans cette phase, l'objectif est de réaliser une analyse approfondie du sujet pour identifier les problèmes, proposer des solutions, et ainsi déduire les exigences fonctionnelles et non fonctionnelles nécessaires à la réalisation du projet.

### Phase 2 : Conception

Une conception solide est essentielle pour garantir le succès du projet. Cette étape implique l'établissement d'un calendrier de réalisation, le choix des plateformes à utiliser, ainsi que la planification des tâches à accomplir.

### **Phase 3 : Développement**

Cette phase se concentre sur la mise en œuvre des fonctionnalités spécifiées dans les exigences. Il s'agit de la phase de codage des éléments à réaliser.

### **Phase 4 : Tests et validation**

Dans cette étape, les fonctionnalités sont testées et tout bug éventuel est corrigé pour garantir le bon fonctionnement du projet.

## **4.4 Planification**

Le diagramme de Gantt présenté ci-dessous illustre la planification des différentes tâches effectuées tout au long de la période de stage.

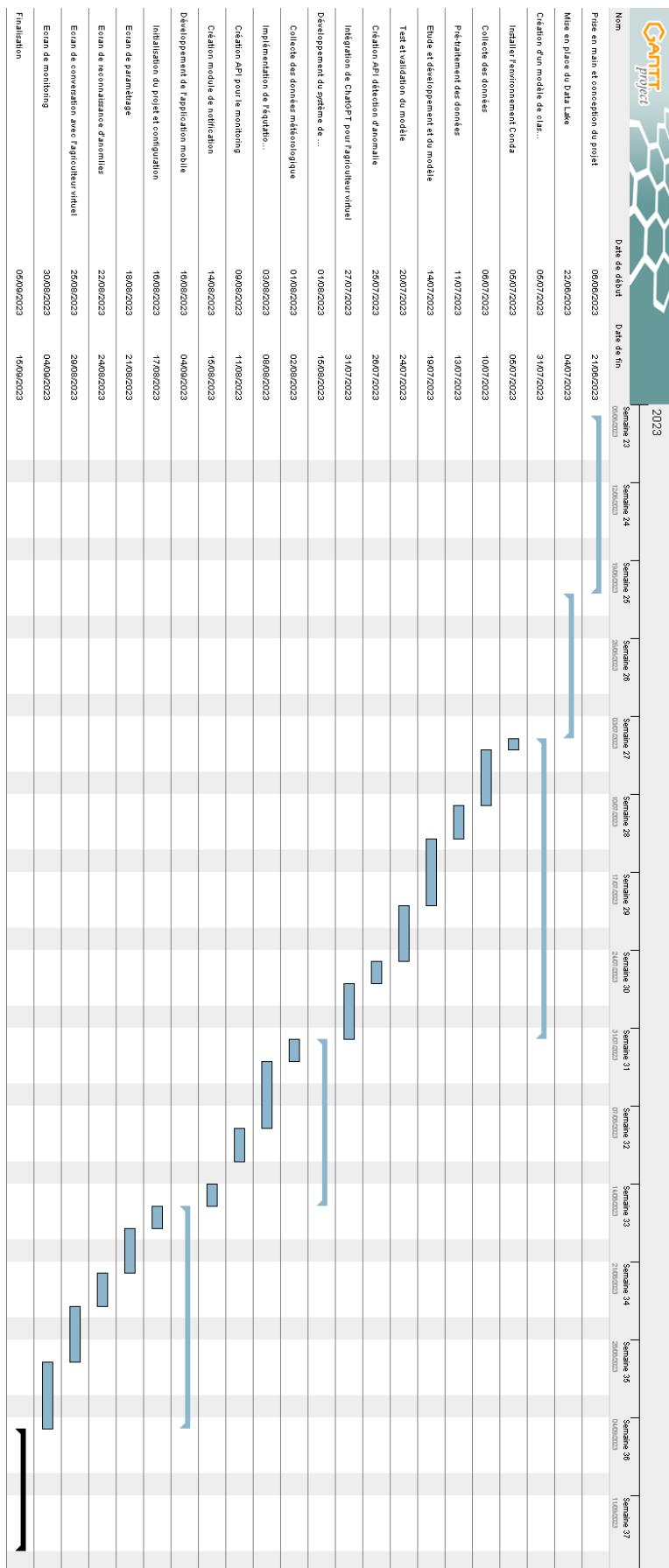


Figure 1. Planning de réalisation

## 4.5 Budget du projet

La mise en place du projet nécessite plusieurs achats de composants tels que les capteurs, les abonnements à des services tiers mais aussi le salaire des développeurs. Notons que certains matériels ou outils comme les ordinateurs portable ont été fourni par l'entreprise d'accueil.

Désignation	Quantité	Coût
Salaire développeur	2	2000000 MGA/Mois/Développeur
Abonnement ChatGPT-4	1	95000 MGA/Mois
Abonnement stockage cloud WASABI	1	32000 MGA/To/Mois
Capteur de niveau d'eau	1/hectare	63000 MGA/Capteur
Capteur d'humidité	1/hectare	300000 MGA/Capteur
Pyranomètre	1	5350000 MGA
Drone - DJI Mini 2 SE	1	1550000 MGA

Tableau 4. Budget du projet

## 5 Exigences réalisées dans le projet (vision externe/utilisateur)

### 5.1 Exigences fonctionnelles – Cas d'utilisation

#### 5.1.1 Détection d'anomalies du riz à partir d'images et suggestion de mesure à prendre

##### Précondition :

- L'application est correctement installée sur le dispositif de l'utilisateur.
- L'utilisateur a accédé à la fonctionnalité de détection d'anomalie.

##### Postcondition :

- Les résultats de la détection d'anomalie sont affichés à l'utilisateur.

##### Déroulement :

- Détection de maladie ou d'anomalie
  - L'application présente à l'utilisateur un menu ou une interface lui permettant de choisir la source des images pour la détection d'anomalie, que ce soit par l'appareil photo intégré, la galerie d'images, ou tout autre moyen pris en charge.
  - L'utilisateur fournit une image (photo) à l'application pour analyse. Le système traite l'image en utilisant le modèle de machine learning pré-entraîné pour la détection d'anomalie sur les feuilles de riz.
  - Le système génère des résultats de détection basés sur l'analyse de l'image, identifiant et classant les anomalies présentes sur la feuille de riz. Les résultats d'analyse sont enregistrés dans le Data Lake.
  - Les résultats de la détection d'anomalie sont présentés à l'utilisateur via l'interface utilisateur, indiquant le type d'anomalie détectée et sa fiabilité.
- Suggestion de mesure à prendre ou discussion avec l'agriculteur virtuel
  - L'utilisateur peut solliciter des conseils sur la manière de résoudre son problème et obtenir des recommandations ou autre.
  - Une interface de conversation s'ouvre, présentant le résultat de la requête.
  - L'utilisateur a la liberté de poser des questions à volonté et de recevoir des réponses de l'application.

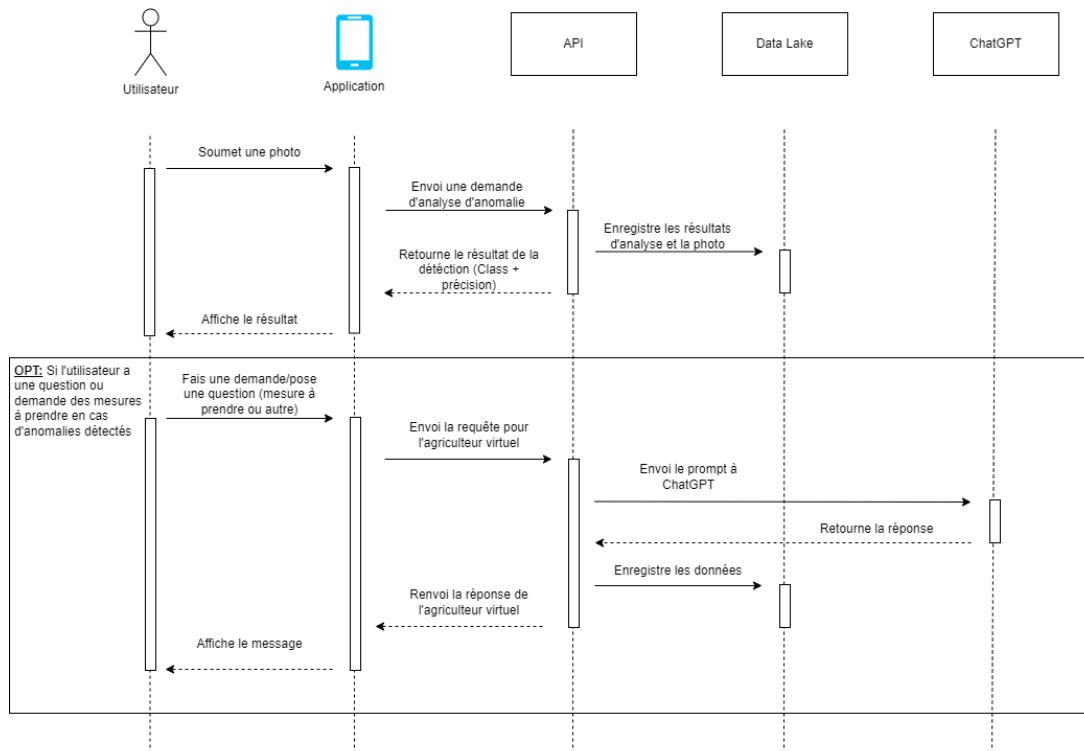


Figure 2. Diagramme de séquence - Détection d'anomalie

### 5.1.2 Monitoring agricole

#### Préconditions :

- L'application est correctement installée et accessible sur le dispositif de l'utilisateur.
- Le système dispose des capteurs nécessaires pour collecter les données agricoles.

#### Postconditions :

- Les données de monitoring sont affichées à l'utilisateur.
- Les informations de monitoring sont stockées dans le Data Lake.

#### Déroulement :

- L'utilisateur lance l'application et accède à la fonctionnalité de monitoring agricole à travers l'interface utilisateur.
- L'application présente à l'utilisateur un tableau de bord ou une interface permettant de visualiser les données agricoles, telles que la température, l'humidité, etc.
- Le système récupère en temps réel les données des capteurs installés sur le terrain, assurant un suivi continu des conditions agricoles.

- Les données collectées sont traitées par le système pour générer des indicateurs de performance, permettant à l'utilisateur de comprendre l'évolution des conditions agricoles.
- Les résultats du monitoring sont présentés à l'utilisateur via l'interface utilisateur.
- L'utilisateur a la possibilité de personnaliser le type de notification et le mode d'envoi qui lui conviennent (Notification via l'application ou notification par SMS) et peut être alerté par le système dans les cas qui auront été implémentés.
- Le système enregistre les informations de monitoring dans le Data Lake, assurant un historique des données agricoles pour une analyse ultérieure.

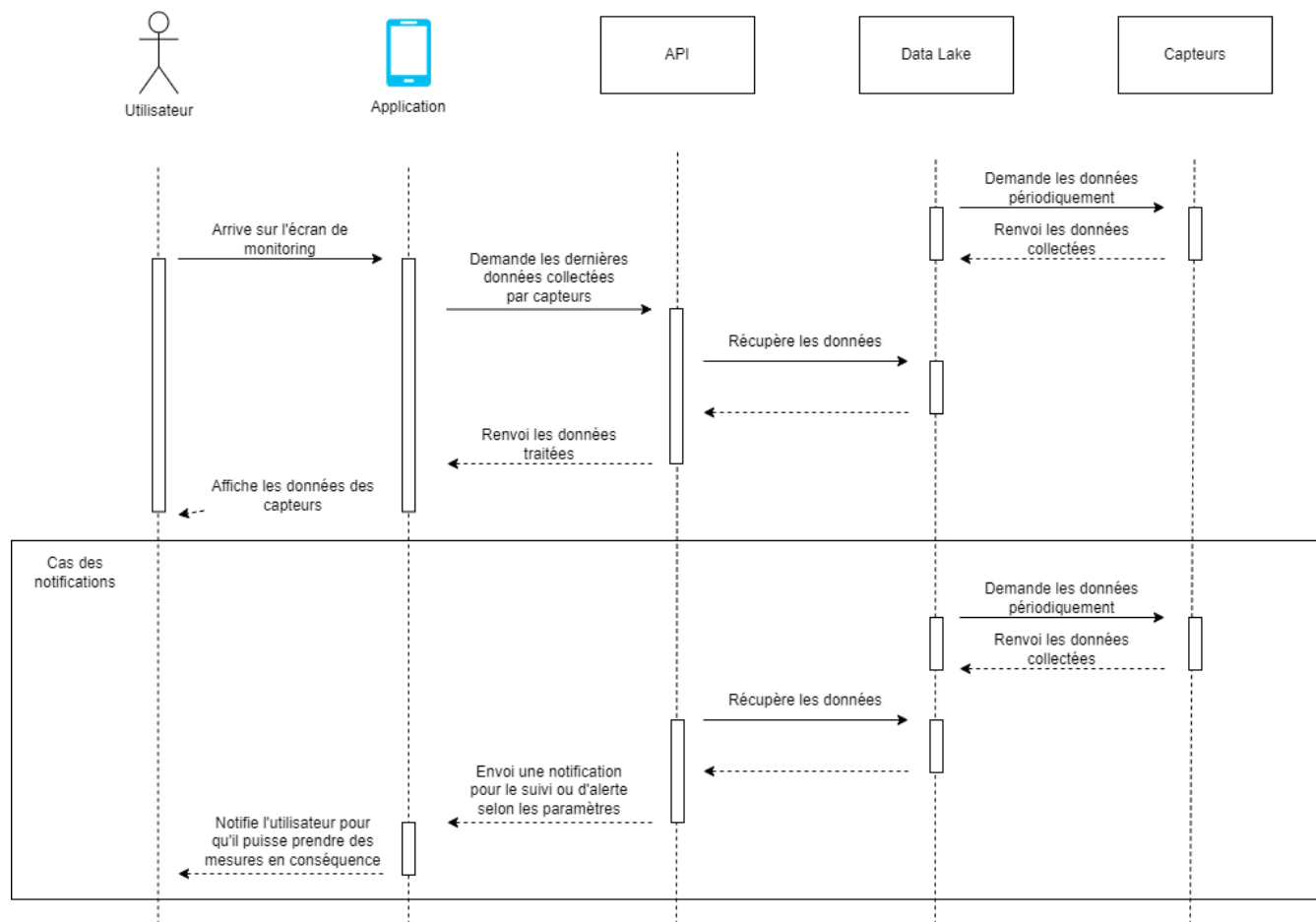


Figure 3. Diagramme de séquence - Monitoring agricole

### 5.1.3 Anticipation d'irrigation

#### Préconditions :

- L'application est installée et accessible sur le dispositif de l'utilisateur.



- Les capteurs nécessaires pour collecter les données météorologiques et du sol sont opérationnels.

Postconditions :

- La prédiction d'irrigation est affichée à l'utilisateur sur l'écran de monitoring.
- Les données de prédiction sont enregistrées dans la base de données.

Déroulement :

- L'utilisateur démarre l'application et accède à la fonctionnalité de prédiction d'irrigation à travers l'interface utilisateur.
- L'application présente à l'utilisateur un tableau de bord ou une interface permettant de visualiser les données météorologiques et du sol actuelles.
- Le système collecte en temps réel les données provenant des capteurs météorologiques et du sol, fournissant des informations précises sur les conditions environnementales.
- Le système utilise l'équation de Penman-Monteith [23] qui estime l'évapotranspiration potentielle pour anticiper les besoins en irrigation en se basant sur les informations qui lui sont données.
- Le système enregistre les informations de prédiction d'irrigation dans la base de données, permettant de constituer un historique pour une analyse ultérieure.
- Les résultats de la prédiction d'irrigation, incluant le moment optimal pour l'irrigation, sont présentés à l'utilisateur via l'interface utilisateur.
- L'utilisateur a la possibilité de personnaliser les paramètres de prédiction, tels que les seuils d'humidité du sol ou les intervalles de prédiction.

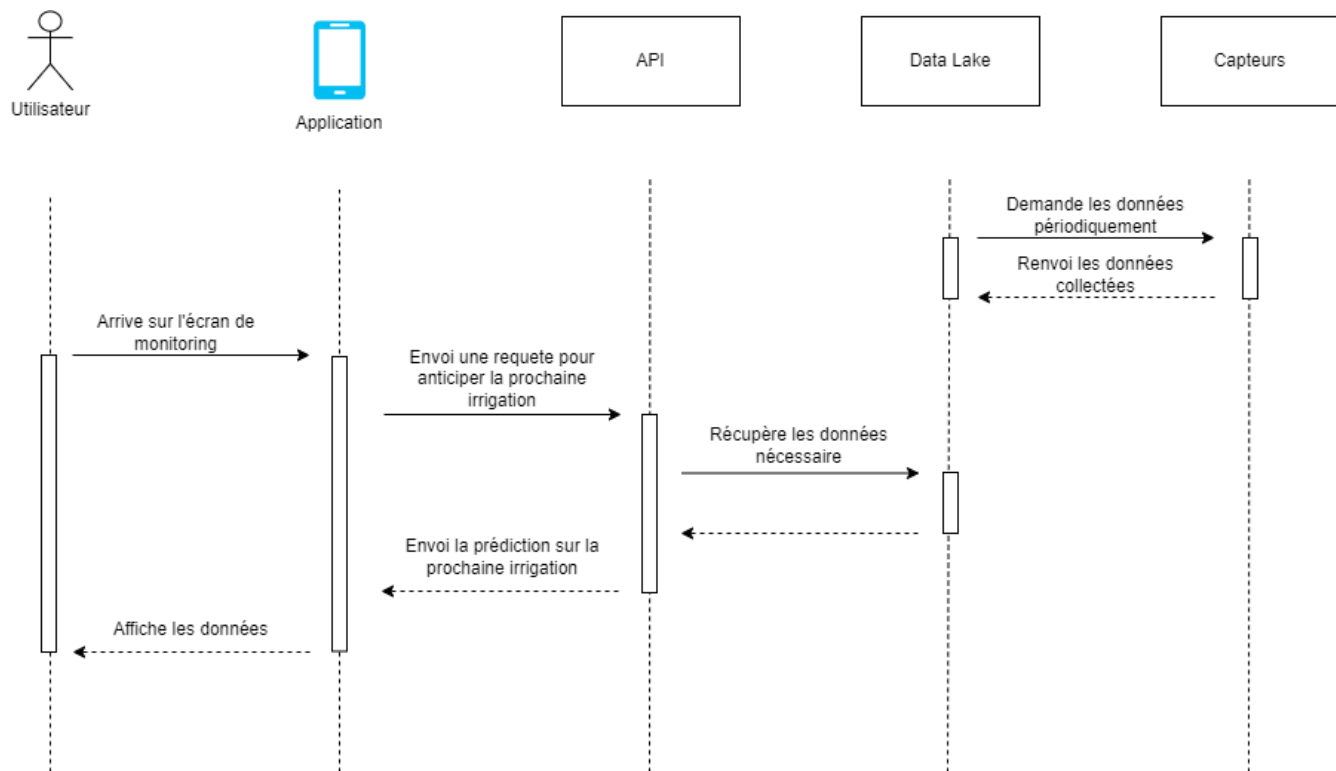


Figure 4. Diagramme de séquence - Anticipation d'irrigation

## 5.2 Exigences non fonctionnelles transverses

### 5.2.1 Mise en place d'un Data Lake

La collecte de données est une étape cruciale dans l'étude de l'agriculture, car la création d'une intelligence artificielle repose sur une abondance de données. Ces données peuvent revêtir diverses formes, telles que structurées, semi-structurées, non structurées, JSON, CSV, images et texte. La qualité des intelligences artificielles créées est étroitement liée à celle des données utilisées.

Afin de gérer efficacement cette diversité de données, la mise en place d'un Data Lake s'avère indispensable. Cette initiative a nécessité l'utilisation d'une machine virtuelle pour installer les différentes sources de données sélectionnées. Ainsi, le Data Lake agit comme un réservoir centralisé permettant de stocker, organiser et gérer ces données hétérogènes, favorisant ainsi la création et l'amélioration d'intelligences artificielles robustes dans le domaine de l'agriculture.

Stockage	Utilisation	Description
Cassandra	Stockage des données semi-	SGBD NoSQL élaboré pour traiter de très

	structuré	gros volume de données distribuées sur de nombreux serveur
Clickhouse	Stockage des données structuré	SGBD orienté colonnes avec une capacité de calculs très rapide, et peut stocker un volume très important de données
HDFS	Stockage des données brutes : CSV, texte, XML	Système de fichiers distribué conçu pour stocker et gérer de très grandes quantités de données sur un cluster de serveurs
Wasabi	Pour stocker les données non structurées comme les photos, vidéos, etc	Service de stockage cloud très performant avec un tarif compétitif

Tableau 5. Conception du Data Lake

### 5.2.2 Performance

L'application que nous avons développée excelle en termes de performance grâce à l'utilisation stratégique de technologies de pointe. L'intégration de Fast API/Uvicorn (en Python) assure une gestion rapide des requêtes, garantissant des temps de réponse courts. Le recours à TensorFlow (en Python) pour le Machine Learning offre une puissance de calcul élevée, assurant une exécution rapide des modèles intégrés. En parallèle, l'utilisation de Flutter (en Dart) pour l'interface utilisateur mobile optimise la réactivité de l'application sur différentes plateformes. Aussi le Data Lake a été conçu de manière à offrir des performances élevées dans le cadre de notre application. Cette combinaison de technologies reflète notre engagement envers des performances élevées, répondant ainsi aux exigences strictes de notre application.

### 5.2.3 Maintenabilité

L'application est considérée comme maintenable du fait de son respect des conventions de codage et celles qui sont propres à chaque technologie, ainsi que des principes de programmation tels que le principe SOLID. Les choix technologiques effectués bénéficient de communautés actives et d'un support continu, garantissant ainsi une adaptabilité aisée de l'application aux évolutions futures.

## 5.3 Interfaces détaillées

### 5.3.1 IHM

#### 5.3.1.1 Cas d'utilisation 1 : Détection d'anomalies du riz à partir d'images et suggestion de mesure à prendre

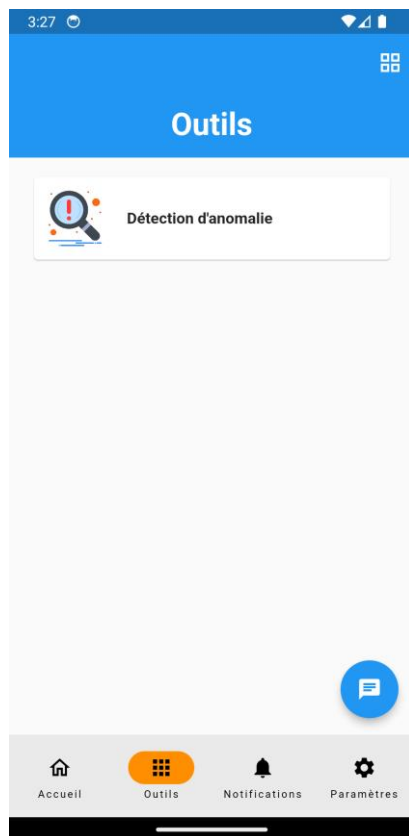


Figure 5. Écran outils

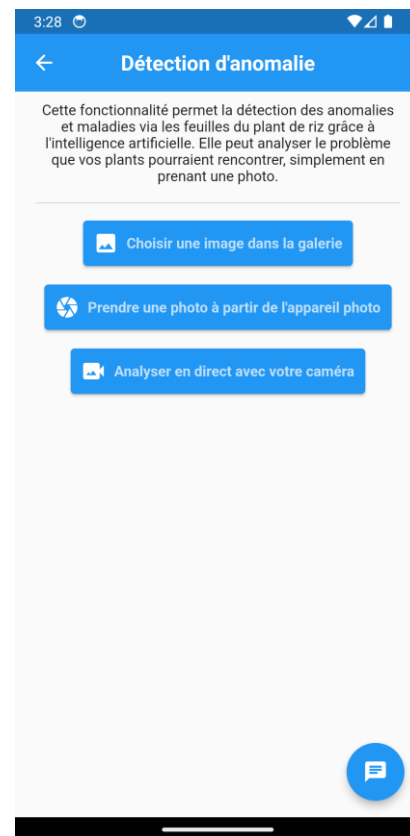


Figure 6. Écran de détection d'anomalies

Sur le premier écran, vous trouverez la liste des fonctionnalités disponibles dans notre application, qui, dans ce contexte spécifique, est axée sur la détection d'anomalies. Lorsque l'utilisateur accède à l'écran de détection d'anomalies, il est incité à sélectionner une image de sa galerie ou à prendre une photo directement avec l'appareil photo de son smartphone. Il est important de noter que la fonction d'analyse en direct avec la caméra est actuellement en cours de développement.

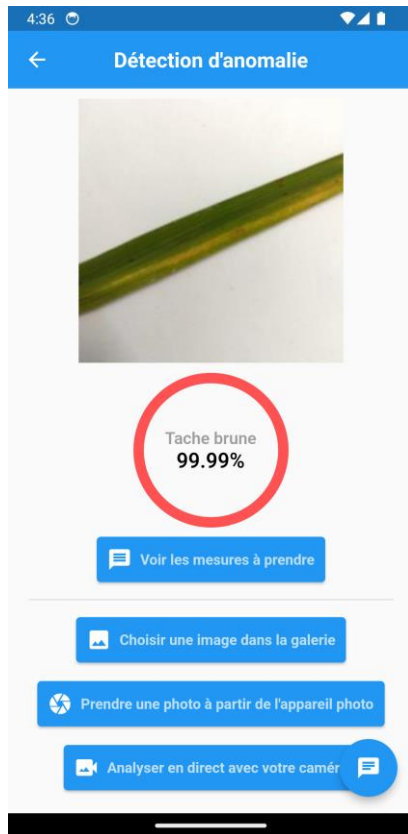


Figure 7. Résultat de la détection d'anomalie

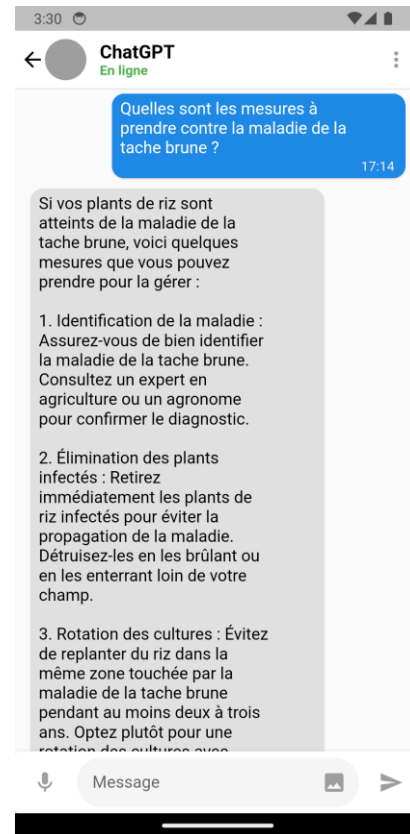


Figure 8. Discussion avec l'agriculteur virtuel

Après avoir soumis la photo, l'écran affiche le résultat de l'analyse, incluant la maladie détectée ainsi que le pourcentage de confiance associé. En dessous de ce résultat, le bouton "Voir les mesures à prendre" permet à l'utilisateur de solliciter l'agriculteur virtuel alimenté par ChatGPT-4 pour obtenir des conseils spécifiques. La conversation avec l'agriculteur virtuel ne se limite pas à cela, et l'utilisateur peut poursuivre en lui posant d'autres questions.

### 5.3.1.2 Cas d'utilisation 2 : Monitoring agricole et anticipation d'irrigation



Figure 9. Ecran d'accueil/Monitoring

Les informations recueillies par les capteurs, les données météorologiques du champ de riz et les prévisions concernant la prochaine irrigation sont affichées sur l'écran d'accueil pour le monitoring.

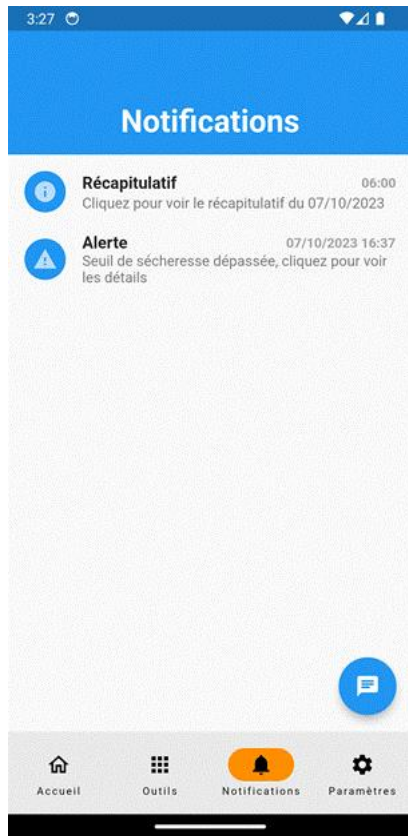


Figure 10. Ecran des notifications

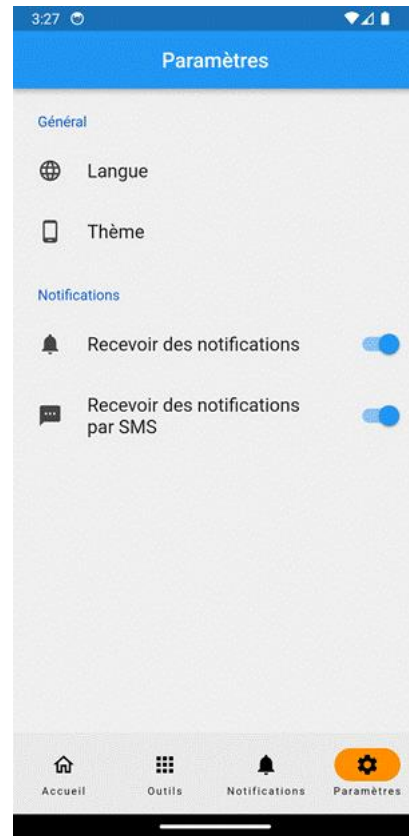


Figure 11. Paramètres de l'application

Sur ces écrans, vous pouvez consulter la liste des notifications transmises à l'utilisateur, accompagnée de quelques paramètres les concernant.

## 5.3.2 Interfaces avec d'autres systèmes

### 5.3.2.1 Interface avec l'API Open-Meteo

En utilisant l'API d'Open-Meteo, nous avons pu obtenir les données météorologiques nécessaires à la création de notre application. Ci-dessous, vous trouverez quelques captures d'écran de l'interface permettant d'utiliser leur service :

open-meteo.com/en/docs#latitude=-18.9137&longitude=47.5361&hourly=temperature\_2m,relative\_humidity\_2m,dew\_point\_2m,precipitation,weather\_code,evapotranspiration,wind\_speed\_10m

## Location and Time

Location: **Coordinates** [List](#)

Latitude: -18,9137 Longitude: 47,5361 Timezone: Not set (GMT+0) [Search](#)

Time: **Forecast Length** [Time Interval](#)

Forecast days: 7 days (default) Past days: 0 (default)

By default, we provide forecasts for 7 days, but you can access forecasts for past weather data, you can use the [Past Days](#) feature to access forecasts.

## Hourly Weather Variables

- ☒ Temperature (2 m)
- ☒ Relative Humidity (2 m)
- ☒ Dewpoint (2 m)
- ☐ Apparent Temperature
- ☐ Precipitation Probability
- ☒ Precipitation (rain + showers + snow)
- ☐ Rain
- ☐ Showers
- ☐ Snowfall
- ☐ Snow Depth
- ☒ Weather code
- ☐ Sealevel Pressure
- ☐ Surface Pressure
- ☐ Cloud cover Total
- ☐ Cloud cover Low
- ☐ Cloud cover Mid
- ☐ Cloud cover High
- ☐ Visibility
- ☒ Evapotranspiration
- ☐ Reference Evapotranspiration (ET<sub>0</sub>)
- ☐ Vapour Pressure Deficit
- ☒ Wind Speed (10 m)
- ☐ Wind Speed (80 m)
- ☐ Wind Speed (120 m)
- ☐ Wind Speed (180 m)
- ☐ Wind Direction (10 m)
- ☐ Wind Direction (80 m)
- ☐ Wind Direction (120 m)
- ☐ Wind Direction (180 m)
- ☐ Wind Gusts (10 m)
- ☐ Temperature (80 m)
- ☐ Temperature (120 m)
- ☐ Temperature (180 m)
- ☐ Soil Temperature (0 cm)
- ☐ Soil Temperature (6 cm)
- ☐ Soil Temperature (18 cm)
- ☐ Soil Temperature (54 cm)
- ☐ Soil Moisture (0-1 cm)
- ☐ Soil Moisture (1-3 cm)
- ☐ Soil Moisture (3-9 cm)
- ☐ Soil Moisture (9-27 cm)
- ☐ Soil Moisture (27-81 cm)

Additional Variables

Solar Radiation Variables

Pressure Level Variables

Weather models

Figure 12. Open-Meteo - Configuration des données météorologique à récupérer

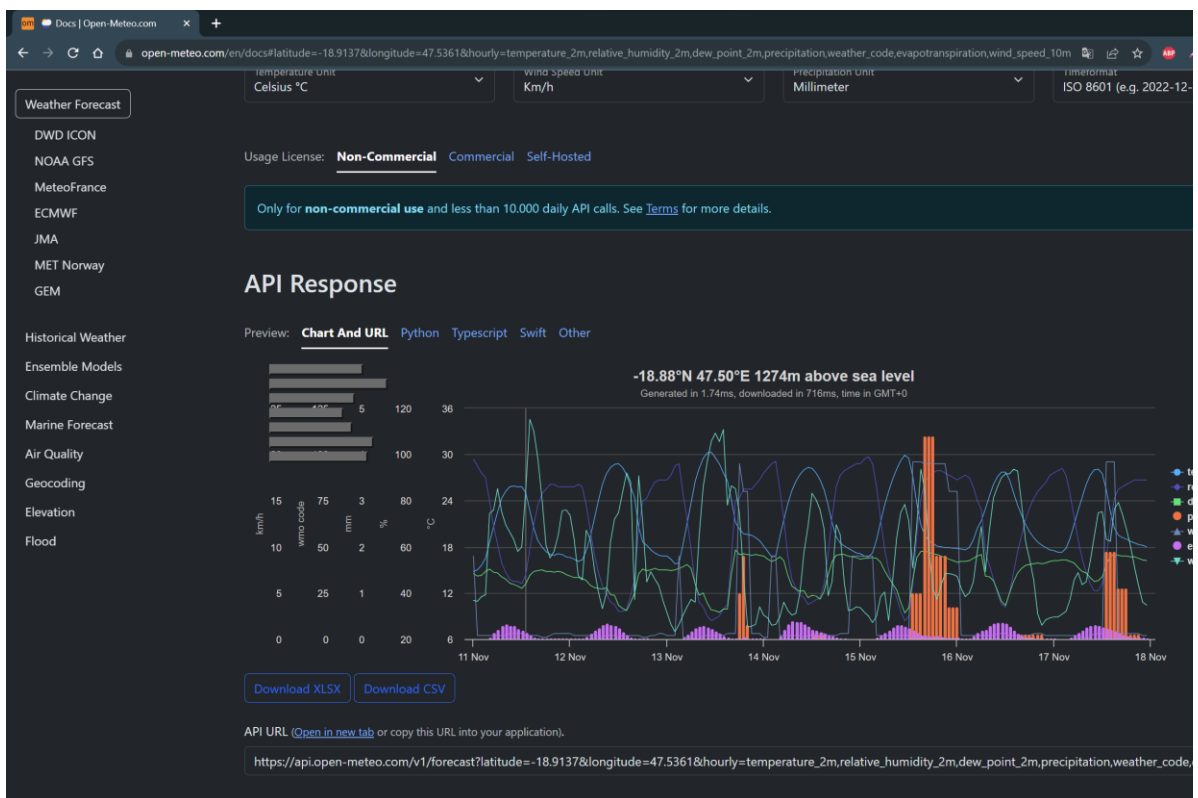


Figure 13. Open-Meteo - Visualisation des données



## 6 Architecture(s) système

Dans cette section, nous allons détailler l'architecture de notre projet, en mettant en lumière l'organisation interne et les éléments matériels qui forment la base de notre système d'agriculture numérique à Madagascar.

### **Architecture Logicielle :**

Dans l'essence de notre solution, les modules logiciels sont les pièces maîtresses orchestrant l'efficacité de notre plateforme.

- Monitoring et anticipation des besoins en irrigation : Il collecte en temps réel des données cruciales grâce aux capteurs intelligents présents dans la rizière et anticipe les besoins en irrigation, en prenant en compte des données météorologiques et des informations des capteurs.
- Gestion Agronomique : L'interface conviviale que nous offrons aux agronomes pour interagir avec les données de la parcelle et recevoir des alertes en cas d'anomalies prend vie dans ce module. Il facilite la prise de décision basée sur des informations actualisées.
- Détection d'Anomalies (CNN) : La détection d'anomalies qui utilise des réseaux neuronaux convolutifs pour analyser les images des feuilles de riz. Il s'agit de notre contribution majeure à la résolution proactive des problèmes agricoles.

### **Architecture Technique :**

Côté matériel, chaque élément joue un rôle crucial dans le fonctionnement harmonieux de notre système.

- Capteurs IoT : Sur le terrain, nos capteurs intelligents, mesurant l'humidité du sol, la température, et bien d'autres paramètres, constituent la première ligne de collecte de données.
- Serveur de Données (Data Lake) : Les données agricoles collectées trouvent leur place dans notre serveur de données dédié, formant ainsi un lac de données robuste.

- Serveur de Traitement : Notre serveur dédié au traitement intensif des données visuelles et à l'apprentissage profond, est le cerveau qui donne vie à nos algorithmes avancés et la logique métier de notre système d'agriculture numérique.
- Interface utilisateur : Le point de contact entre un utilisateur et le serveur de traitement. C'est la plateforme visuelle et interactive à travers laquelle les utilisateurs interagissent avec notre système.

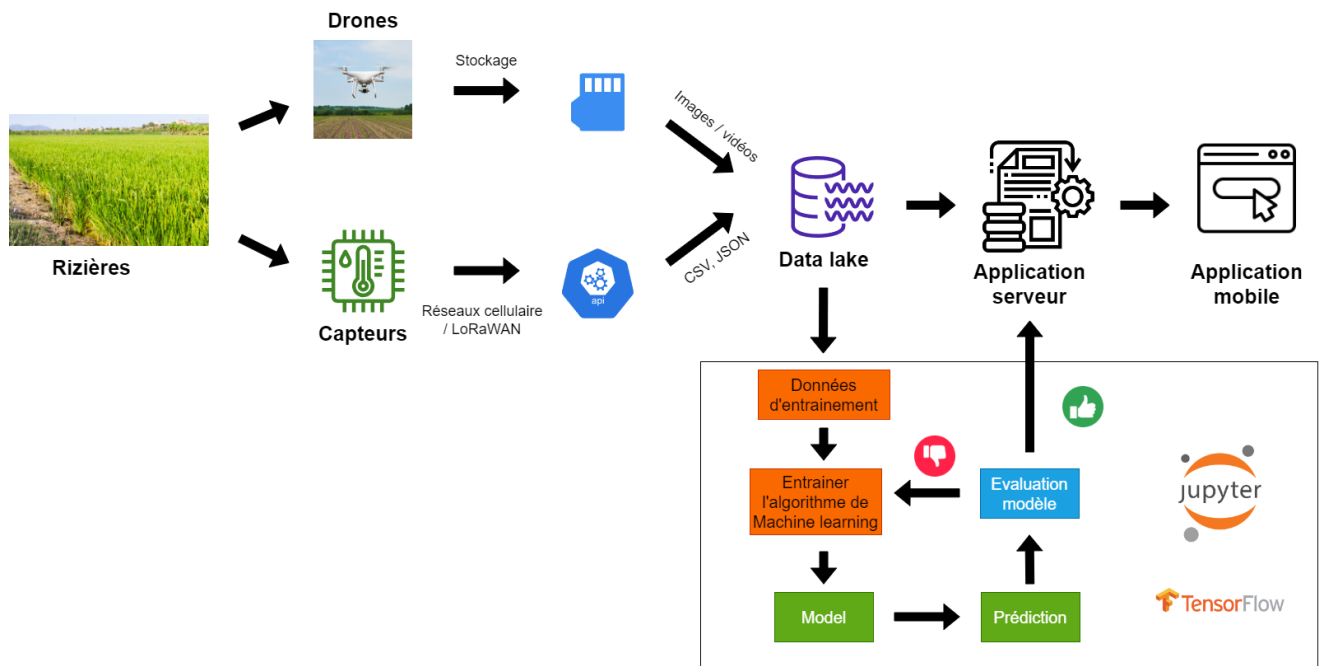


Figure 14. Architecture du système

## 7 Conception du système logiciel réalisée dans le projet (vision interne/développeur)

### 7.1 Plate-forme technique

Dans le processus de conception d'un logiciel, la sélection de la technologie revêt une importance particulière. Il est essentiel de choisir une technologie qui réponde adéquatement aux exigences du logiciel tout en tenant compte des contraintes spécifiques du projet. Dans cette optique, examinons maintenant les frameworks et bibliothèques que nous avons choisies pour mettre en œuvre notre application :

- **Fast API**
  - Version : 0.103.1

- Langage : Python
- Utilisation : Framework utilisé pour développer l'application serveur.
- Justification : Elle fournit des performances plus élevées que certaines Framework comme Node.js, facile d'utilisation, génère automatique la documentation de l'API et est adapté aux applications de data science.
- **Tensorflow**
  - Version : 2.9.3
  - Langage : Python
  - Utilisation : Bibliothèque utilisée pour développer les modèles d'apprentissage automatique.
  - Justification : Performances élevées, facile d'utilisation, large écosystème de fonctionnalité et support communautaire.
- **Flutter**
  - Version : 3.13.6
  - Langage : Dart
  - Utilisation : Framework utilisé pour développer l'application mobile.
  - Justification : Performances élevées et facilité d'utilisation, développement multiplateforme natif qui permet de réduire les coûts de développement et de maintenance, large communauté grandissante, bien documentée.

Dans la conception de notre application d'agriculture numérique, la sélection de l'architecture matérielle joue un rôle déterminant. Il est crucial de choisir une infrastructure qui réponde de manière optimale aux besoins spécifiques de notre application tout en tenant compte des contraintes inhérentes au domaine agricole. Dans cette perspective, explorons à présent les composants matériels, en particulier les capteurs, que nous avons judicieusement choisis pour mettre en œuvre notre solution :

- **Capteurs niveau d'eau** : pour vérifier le niveau d'immersion du riz
- **Capteurs humidité** : pour avoir le taux d'humidité du sol et hors sol, ainsi que la température

- **Pyranomètre** : pour mesurer l'intensité de rayonnement du soleil

## 7.2 Conception du logiciel développé

### 7.2.1 Conception du code source

Dans le développement de notre application, nous avons organisé le code source en trois dépôts distincts, correspondant à chaque composante de notre système : le machine learning, le backend, et le frontend. La structure des code sources a été pensée de sorte qu'elle soit compréhensible et maintenable. Chacune de ces sections adopte des technologies spécifiques ayant chacune une architecture qui leur est propre et nous allons désormais détailler cela.

#### **Machine learning :**

Dans cette partie, vous trouverez des fichiers Jupyter Notebook renfermant le code source des divers modèles de classification que nous avons élaborés et son répertoire est organisé de la manière suivante :

- Exports : Englobe les diverses versions des modèles de classification que nous aurons générés.
- Notebooks : Regroupe les fichiers dédiés à la création des modèles.

#### **Backend :**

Le répertoire Backend, qui héberge un projet en Python utilisant le framework Fast API, suit une convention de nommage en Snake case, où les mots sont en minuscules et séparés par un tiret bas. L'organisation de ce répertoire est la suivante :

- Api : Inclut chaque version de l'API, comprenant les routeurs et les contrôleurs.
- Models : Rassemble les classes d'objets représentant les entités en base de données.
- Services : Héberge la logique métier des différents modules de l'application.
- Schemas : Stocke les modèles de validation de données utilisés par l'API dans les requêtes et les réponses.

#### **Frontend :**

Le dépôt frontend, qui englobe notre application mobile développée avec le framework Flutter et suit également la convention de nommage en Snake case, est organisé selon la structure suivante :

- **Components** : Rassemble les différents widgets réutilisables.
- **Helpers** : Contient les utilitaires que nous utiliserons dans toute l'application.
- **L10n** : Stocke les données de traduction des différentes langues disponible dans l'application.
- **Models** : Héberge les modèles de données.
- **Pages** : Contient les écrans ou vues principales de l'application.
- **Providers** : Contient des classes qui fournissent des données à d'autres parties d'une application. Ils sont utilisés pour implémenter le modèle-vue-contrôleur (MVC) dans Flutter.
- **Services** : Héberge la logique métier des différents modules.
- **Views** : Contient les différents écrans ou vues secondaire de l'application.

## **7.2.2 Conception du modèle de classification d'anomalies Deep Learning**

### **7.2.2.1 Images des anomalies du riz**

Pour entraîner notre modèle, nous avons employé un jeu de données extrait du site Kaggle [24], lequel comporte quatre catégories distinctes associées à des feuilles de riz. Parmi celles-ci, trois catégories représentent des maladies, tandis que la quatrième représente des feuilles saines. Les quatre classes sur lesquelles nous allons focaliser notre travail sont les suivantes :

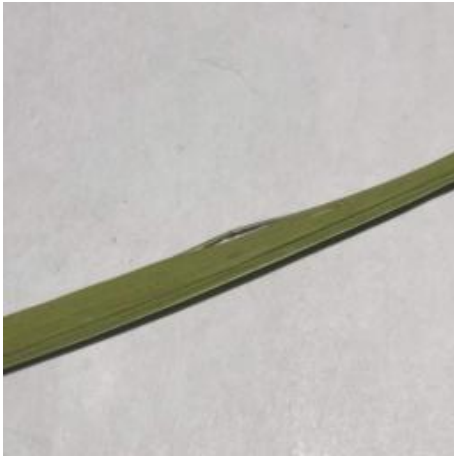


Figure 15. Feuille saine



Figure 16. Tache brune

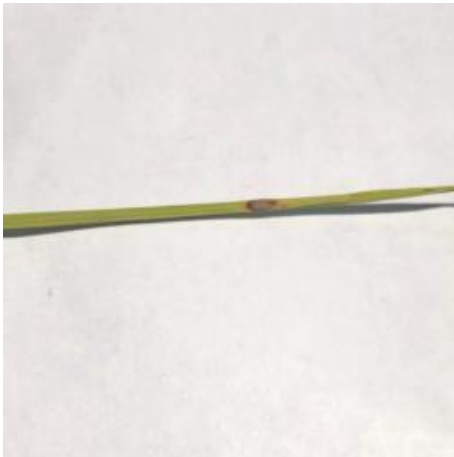


Figure 17. Rouille des feuilles



Figure 18. Blast du riz

Notre ensemble de données compte 4081 images qui a été divisé comme suit :

- 3265 images pour l'entraînement
- 816 images pour la validation du modèle

Afin d'avoir une meilleure idée sur le modèle approprié à notre application, nous allons comparer 3 modèles de machine learning basé sur la bibliothèque Tensorflow :

- Modèle CNN standard
- Modèle basé sur le modèle Xception
- Modèle basé sur le modèle MobileNetV2

### 7.2.2.2 Modèle CNN standard

Le modèle séquentiel est composé de six ensembles de blocs de convolution (Conv2D) utilisant la fonction d'activation ReLU, chaque bloc étant suivi d'une couche de regroupement maximale (MaxPooling2D) de dimension (2, 2). Une couche d'aplatissement (Flatten) est ensuite intégrée pour transformer les données en un vecteur unidimensionnel, préparant ainsi l'entrée pour les couches entièrement connectées. Les dernières couches entièrement connectées (Dense) comportent 64 unités chacune, et le nombre de neurones dans la dernière couche correspond au nombre de classes, générant des probabilités associées à chaque classe. Ces dernières couches sont activées respectivement par les fonctions ReLU et Softmax.

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 4

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])

model.build(input_shape=input_shape)
```

Figure 19. Modèle séquentiel

```
scores = model.evaluate(test_ds)

77/77 [=====] - 2s 21ms/step - loss: 0.1501 - accuracy: 0.9557
```

Figure 20. Evaluation: Modèle basé sur le modèle séquentiel

### 7.2.2.3 Modèle basé sur le modèle Xception

Xception, basée sur des convolutions séparables en profondeur, représente une nouvelle architecture de réseau neuronal convolutif profond qui s'inspire du modèle Inception.

Dans Xception, les modules Inception sont remplacés par des convolutions séparables en profondeur suivies de convolutions ponctuelles. Cette architecture efficiente repose sur deux aspects fondamentaux:

- Convolution en profondeur (Depthwise convolution) : Il s'agit d'une convolution effectuée indépendamment sur chaque canal d'entrée.
- Convolution ponctuelle (Pointwise convolution): Cette convolution 1x1 projette les canaux après la convolution en profondeur vers un nouvel espace de canaux.

La convolution en profondeur applique un filtre à chaque canal d'entrée, tandis que la convolution ponctuelle effectue une convolution 1×1 pour fusionner les sorties de la convolution en profondeur. Cette fusion efficace est rendue possible grâce à la mise en œuvre performante de la convolution en profondeur proposée par TensorFlow.

Comparée à la convolution standard, qui filtre et combine les données d'entrée en une seule étape, la convolution en profondeur séparable décompose ce processus en deux couches distinctes : une couche dédiée au filtrage et une autre à la fusion.

```
base_model = tf.keras.applications.Xception(input_shape = IMG_SHAPE, include_top = False, weights = 'imagenet')

base_model.trainable = False

out = base_model.output
out = tf.keras.layers.GlobalAveragePooling2D()(out)
out = tf.keras.layers.Flatten()(out)
out = tf.keras.layers.Dense(32, activation = 'relu')(out)
out = tf.keras.layers.Dense(16, activation = 'relu')(out)
predictions = tf.keras.layers.Dense(len(class_names), activation = 'softmax')(out)
model = tf.keras.Model(inputs = base_model.input, outputs = predictions)
```

Figure 21. Modèle basé sur le modèle Xception

```
loss0, accuracy0 = model.evaluate(val, steps = validation_steps)

20/20 [=====] - 2s 102ms/step - loss: 0.1737 - accuracy: 0.8095
```

Figure 22. Evaluation: Modèle basé sur le modèle Xception

#### 7.2.2.4 Modèle basé sur le modèle MobileNetV2

En 2017, Google a intégré les architectures pré-entraînées de MobileNet à TensorFlow, en suivant des principes similaires à ceux de Xception, afin de rendre les modèles encore plus compacts. Ces modèles sont conçus pour exécuter des tâches de vision par ordinateur directement sur des dispositifs mobiles ou des objets connectés, caractérisés par des limitations importantes en termes de mémoire et de puissance de traitement. Cette particularité s'avère particulièrement pertinente pour notre projet.



MobileNet utilise des convolutions séparables en profondeur de dimension  $3 \times 3$ , permettant ainsi de réduire le nombre de calculs de 8 à 9 fois par rapport aux convolutions standard, tout en minimisant la perte de précision.

```
base_model = tf.keras.applications.MobileNetV2(input_shape = IMG_SHAPE, include_top = False, weights = 'imagenet')

base_model.trainable = False

out = base_model.output
out = tf.keras.layers.GlobalAveragePooling2D()(out)
out = tf.keras.layers.Flatten()(out)
out = tf.keras.layers.Dense(512, activation = 'relu')(out)
out = tf.keras.layers.Dropout(0.5)(out)
out = tf.keras.layers.Dense(512, activation = 'relu')(out)
out = tf.keras.layers.Dropout(0.5)(out)
predictions = tf.keras.layers.Dense(len(class_names), activation = 'softmax')(out)
model = tf.keras.Model(inputs = base_model.input, outputs = predictions)
```

Figure 23. Modèle basé sur le modèle MobileNetV2

```
loss0, accuracy0 = model.evaluate(val, steps = validation_steps)

20/20 [=====] - 1s 60ms/step - loss: 0.4403 - accuracy: 0.8125
```

#### 7.2.2.5 Conclusion

Dans notre cas, le modèle standard CNN a montré le meilleur résultat en termes de précision de prédiction et de vitesse de calcul.

	Modèle standard CNN	Modèle basé sur le modèle Xception	Modèle basé sur le modèle MobileNetv2
Précision, %	97%	80%	81%
Performance, ms/step (GPU)	21ms/step	102ms/step	60ms/step

Tableau 3. Résultat en termes de précision de prédiction et de vitesse de calcul

### 7.2.3 Le code source – vue statique

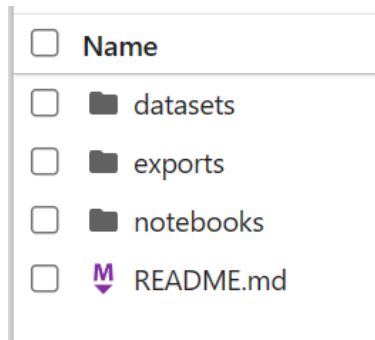


Figure 24. Structure code source:  
Machine learning

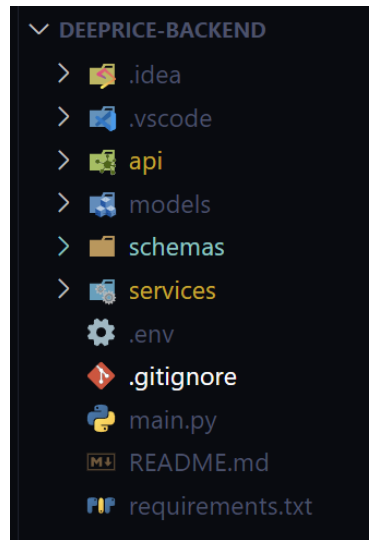


Figure 25. Structure code source:  
Backend

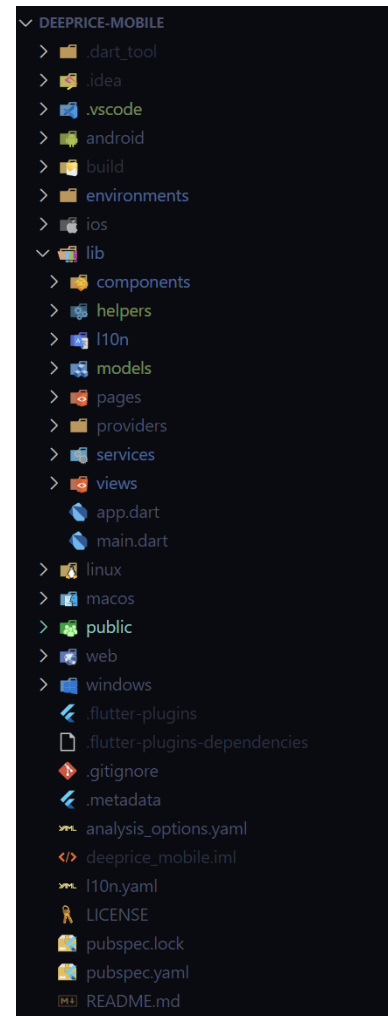


Figure 26. Structure code source:  
Frontend (application mobile)

### 7.2.4 Les composants et leur déploiement

En ce qui concerne le déploiement, l'application mobile peut être mise en service sur les diverses boutiques d'applications en fonction de la plateforme, ou simplement installer le fichier APK construit dans le cas d'une plateforme Android. En ce qui concerne l'application serveur, qui abrite le cœur et la logique métier de notre application, elle sera déployée sur un serveur dédié, intégrant le modèle de classification des anomalies des plants de riz que nous aurons entraîné.

## 8 Tests du système logiciel

Chaque développeur effectue des tests unitaires et des tests en local avant de soumettre une "pull request" une fois qu'une fonctionnalité est terminée. Son code est ensuite passé en revue et validé par le gestionnaire ou un autre développeur de l'équipe. Des tests d'intégration sont ensuite réalisés sur un serveur de développement. Des corrections peuvent être apportées avant les tests de validation dirigés par le chef de projet. Les développeurs peuvent également être appelés à tester plusieurs parties du projet.

## 9 Conclusion générale

En conclusion de ce stage, le développement de notre application a atteint des étapes significatives, permettant aux utilisateurs de détecter des anomalies, surveiller l'état de leurs rizières, et recevoir des prédictions d'irrigation. Ces réalisations, composées respectivement de 3933 et 378 lignes de code pour l'application mobile et l'application serveur, démontrent une avancée significative.

Actuellement, le développement de l'application est toujours en cours, mais nous pouvons dire que la majorité des objectifs ou module demandés pour cette première version ont été accomplis. La fonctionnalité de monitoring est opérationnelle, bien qu'elle ne reçoive pas encore de données en temps réel en raison de l'absence actuelle des capteurs. De même, la prévision de l'irrigation suivante a été implémentée, capable de prédire cette dernière en fonction des données fournies. La reconnaissance d'anomalies est fonctionnelle et a été formée sur trois types d'états/anomalies du riz, affichant une précision dépassant les 95% dans la plupart des cas. Le système d'agriculture numérique basé sur ChatGPT-4 répond actuellement aux besoins spécifiques.

Au cours de la réalisation du projet, la collecte de données a présenté des défis, notamment le manque de données exploitables dans le domaine de la plantation de riz à Madagascar. La saison de plantation d'octobre à mars a également limité la collecte de données sur le terrain. Pour surmonter ces obstacles, nous avons opté pour la récupération de données provenant d'autres pays sur le web, contribuant ainsi au progrès de nos modèles d'apprentissage automatique. Pour l'anticipation d'irrigation, il a été nécessaire d'utiliser l'équation de Penman-Monteith due à l'absence de données.

Dans la perspective de l'avenir de l'agriculture numérique, l'intégration de capteurs et l'utilisation optimale des données collectées représentent des axes d'amélioration. L'introduction d'un suivi par parcelle permettrait aux agriculteurs d'enregistrer divers paramètres, de la nutrition du sol au rendement des récoltes. Ces données pourraient ensuite servir de base à de nouvelles fonctionnalités, telles que la prédiction du rendement ou la proposition de traitements optimisés.

Sur le plan personnel, ce stage a été une opportunité enrichissante pour explorer la réalité agricole à Madagascar et découvrir les vastes possibilités offertes par l'IA et l'IoT, tant dans l'agriculture que dans d'autres secteurs. Cette expérience a approfondi ma compréhension et a renforcé mes compétences pratiques dans le domaine passionnant de l'IA et de l'IoT.

## 10 Références et Bibliographie

- [1] Travail inspirée du mémoire en doctorat du docteur Aygyul EPIMAKHOVA, IoT et IA (Tensorflow/COLAB) en Agriculture numérique appliquée à la vigne, septembre 2022.
- [2] Perspectives économiques à Madagascar : une croissance soutenue mais qui doit bénéficier davantage aux plus pauvres. Juillet 2018.  
<https://www.banquemonddiale.org/fr/country/madagascar/publication/madagascar-economic-update-fostering-financial-inclusion>
- [3] Consommation de riz à Madagascar : 151.000 tonnes de riz importé en janvier et février 2023. 13 mai 2023. <https://www.madagascar-tribune.com/Consommation-151-000-tonnes-de-riz-importe-en-janvier-et-fevrier.html>
- [4] Sans eau, pas d'agriculture. <https://www.fert.fr/sans-eau-pas-dagriculture-les-paysans-malgaches-se-mobilisent/>
- [5] Qu'est-ce que l'intelligence artificielle? <https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/>
- [6] Qu'est-ce que l'Internet des objets (IoT) ?  
<https://www.sap.com/suisse/products/artificial-intelligence/what-is-iot.html>
- [7] Qu'est-ce qu'un Data Lake? Disponible sur : <https://www.oracle.com/fr/database/data-lake-definition/>
- [8] Abhishek Sharma (Février 2022) How to Install TensorFlow with Cuda and cuDNN support in Windows. 26 février 2022. <https://machinelearningprojects.net/install-tensorflow-with-cuda-and-cudnn/>
- [9] Saturn Cloud, How to Run Jupyter Notebook on GPUs par Saturn Cloud. 1 juin 2023.  
<https://saturncloud.io/blog/how-to-run-jupyter-notebook-on-gpus/>
- [10] Image Classification Using CNN: Introduction and Tutorial.  
<https://datagen.tech/guides/image-classification/image-classification-using-cnn/>
- [11] Image Classification Using CNN. 14 juin 2023.  
<https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/>

- [12] Ilyes Talbi, Tutoriel : utiliser l'API ChatGPT avec Python. 2 mars 2023.  
<https://larevueia.fr/tutoriel-utiliser-lapi-chatgpt-avec-python/>
- [13] Install Hadoop on Ubuntu. 17 mars 2023. <https://learnubuntu.com/install-hadoop/>
- [14] How to Install ClickHouse on Ubuntu 22.04 LTS Linux. 16 novembre 2022.  
<https://linux.how2shout.com/how-to-install-clickhouse-on-ubuntu-22-04-lts-linux/>
- [15] Installing Cassandra | Apache.  
[https://cassandra.apache.org/doc/latest/cassandra/getting\\_started/installing.html#installing-the-debian-packages](https://cassandra.apache.org/doc/latest/cassandra/getting_started/installing.html#installing-the-debian-packages)
- [16] Fast, Affordable Cloud Storage & Secure Data Protection – Wasabi.  
<https://wasabi.com/>
- [17] Comment installer Ubuntu sur VirtualBox? 2 juillet 2023.  
<https://www.minitool.com/fr/disque-partition/comment-installer-ubuntu-sur-virtualbox.html>
- [19] FASAL. <https://fasal.co/>
- [20] Phytech. <https://www.phytech.com/>
- [21] Banque mondiale, “WP#101 Rice Markets in Madagascar - World Bank”.  
<https://documents1.worldbank.org/curated/en/998501468056668263/pdf/387060PAPER0MG1rkets0wp10101PUBLIC1.pdf>
- [22] Starlink. <https://www.starlink.com/>
- [23] Formule de Penman.  
<https://www.aquaportail.com/dictionnaire/definition/6449/formule-de-penman>
- [24] Données sur Kaggle. <https://www.kaggle.com/datasets/nafishamoin/new-bangladeshi-crop-disease>

# 11 Annexes

## 11.1 Annexe 1 : Interface interaction du Data Lake dans la machine virtuelle

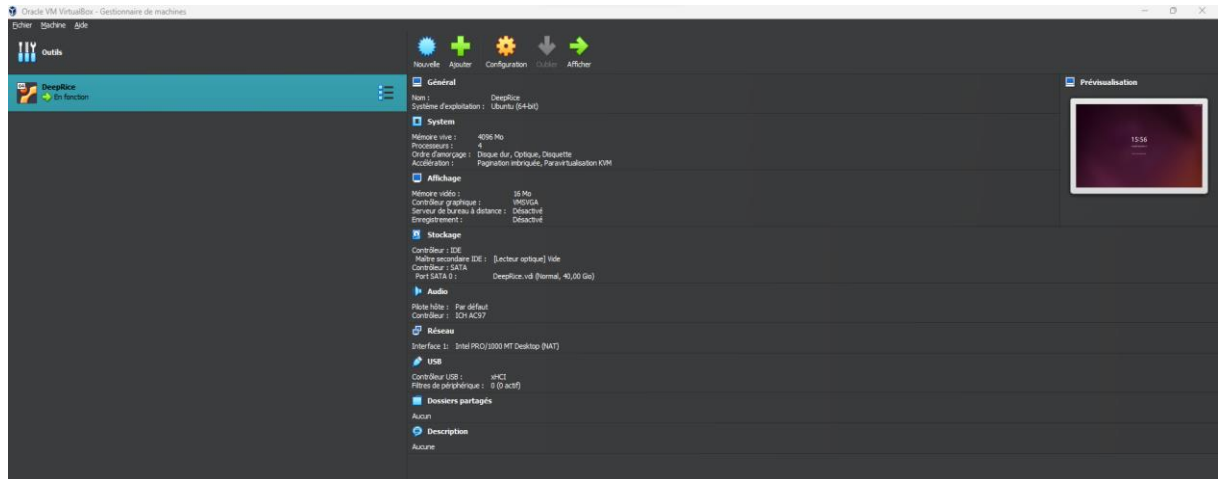


Figure 27. Liste de machine virtuelle dans virtualBox

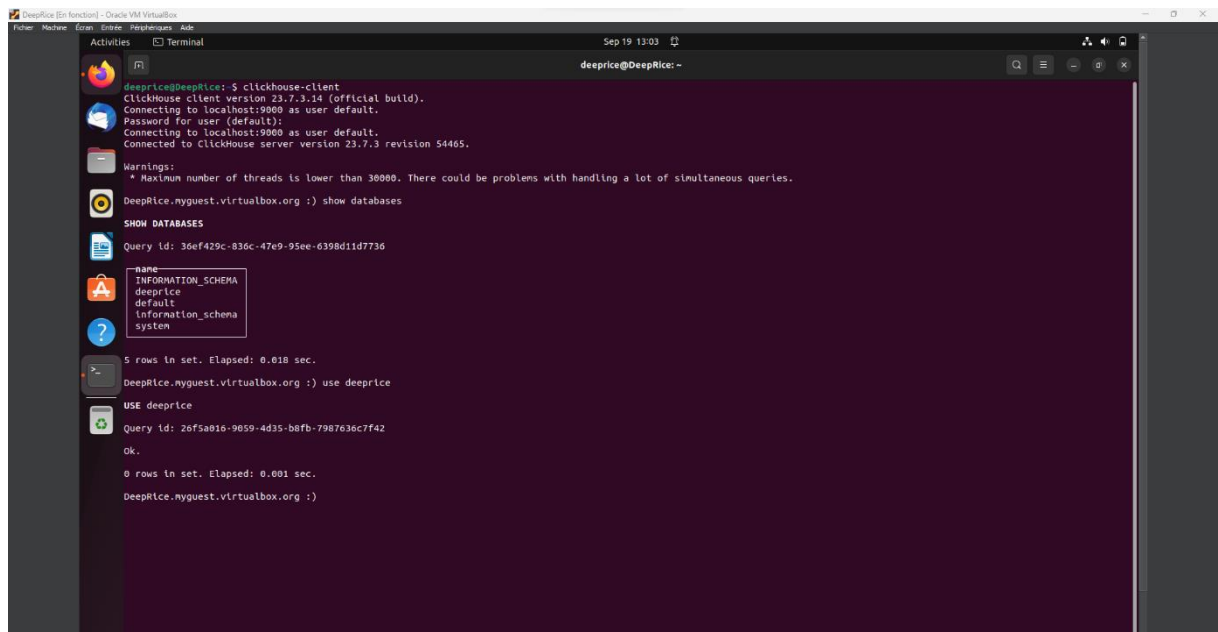


Figure 28. Connexion à clickhouse

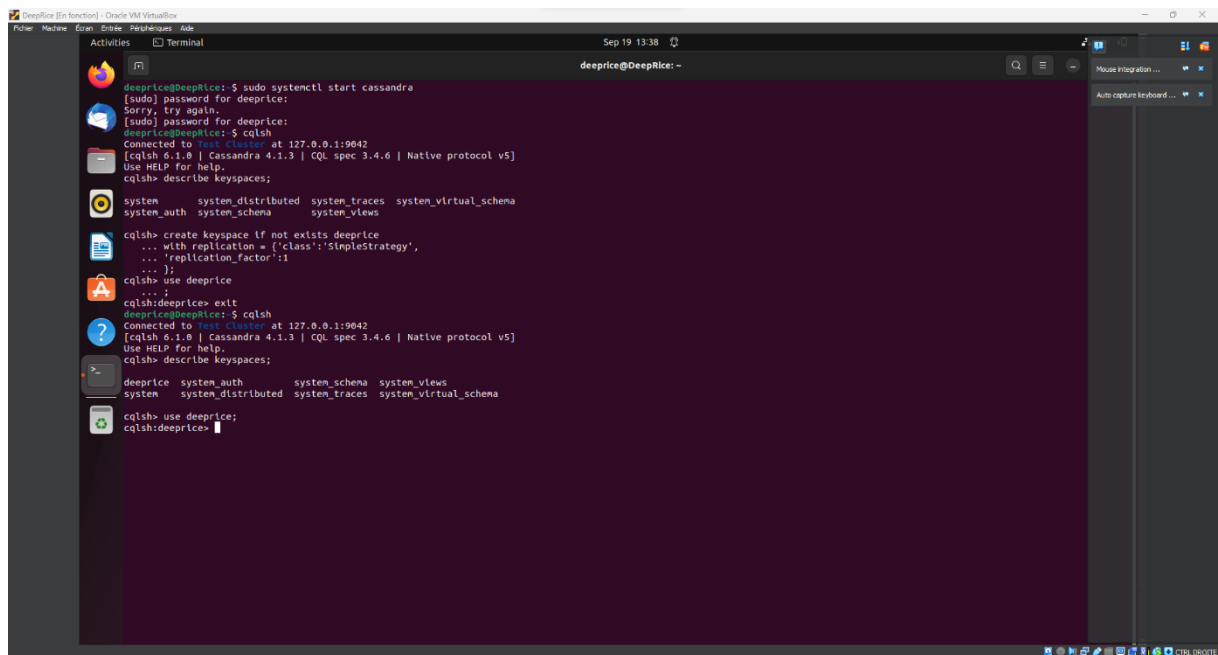


Figure 29: Connexion avec Cassandra

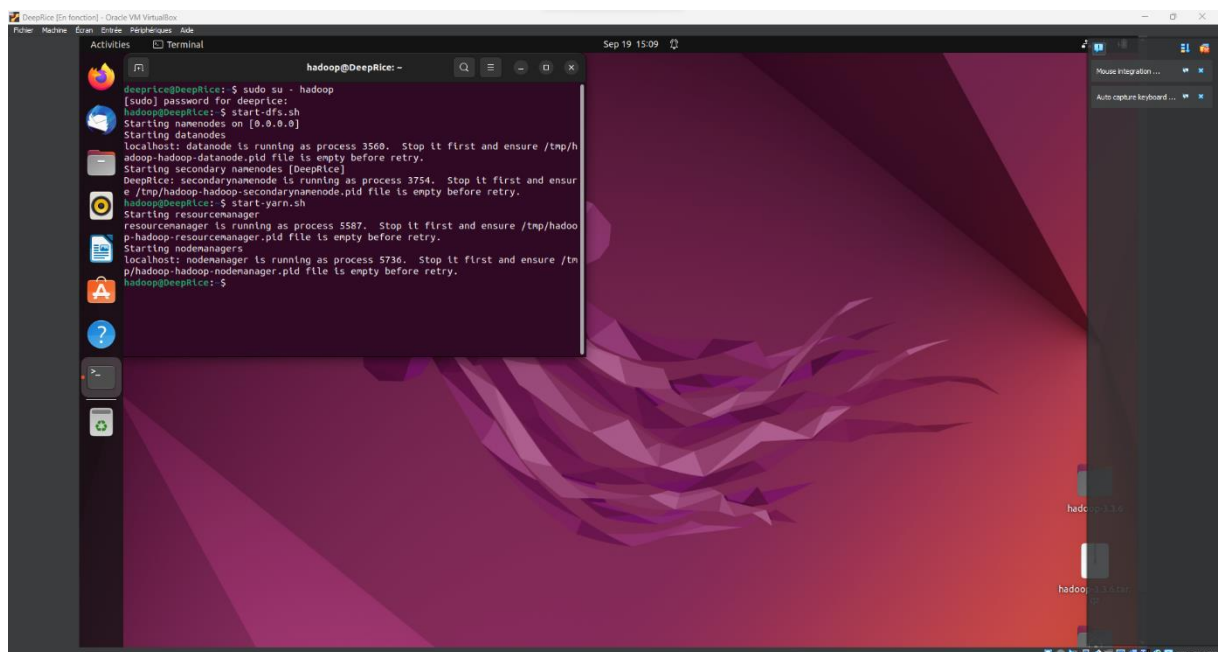


Figure 30. Démarrage de HDFS

## 11.2 Annexe 2 : Comparaison des sources de données

Critères	Google Drive	Wasabi
Type	Stockage Cloud	Stockage Cloud



Tarification	Offre une version gratuite avec limites de stockage. Des forfaits payants disponibles avec plus d'espace.	Facturation basée sur la capacité de stockage utilisée, avec des tarifs fixes.
Capacité de stockage	Jusqu'à 15 Go gratuits (avec possibilité d'extension via abonnements payants).	Pas de limite supérieure sur la capacité de stockage.
Performance	Convient aux petits fichiers et à un usage personnel. Peut-être plus lent pour les transferts de gros fichiers.	Conçu pour une haute performance, y compris le transfert de gros fichiers et l'accès rapide aux données.
Stockage à long terme	Non spécifiquement conçu pour le stockage à long terme.	Conçu pour le stockage à long terme avec des coûts prévisibles.
Support technique	Assistance via la documentation en ligne, le support communautaire et le support payant pour les utilisateurs professionnels.	Support client dédié, avec options de support payant.

Tableau 6. Comparaison entre Google drive et Wasabi

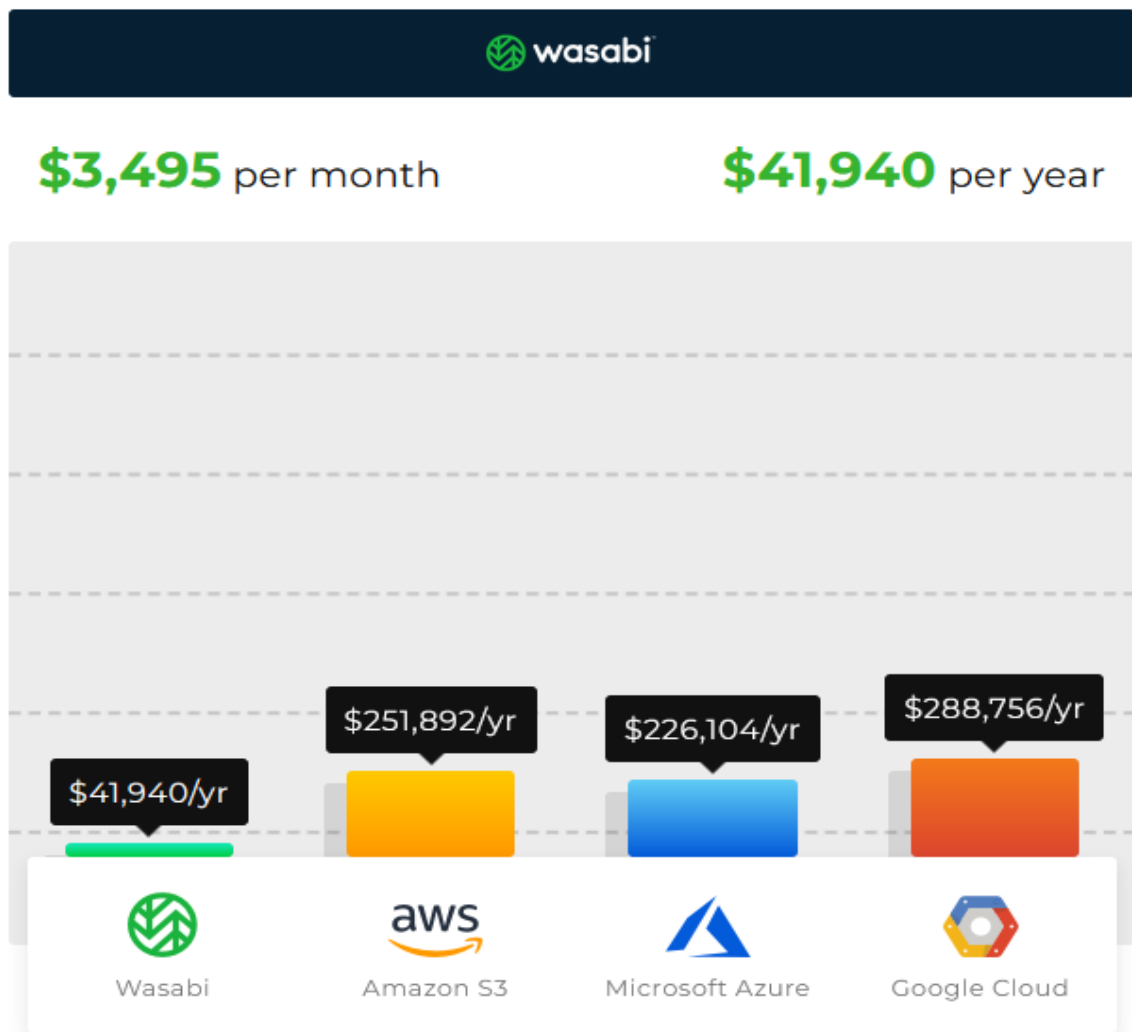


Figure 31. Comparaison prix sur les stockages cloud

Caractéristique	HDFS	ClickHouse	Cassandra
Type de système de stockage	Système de fichiers distribué	Système de gestion de base de données orienté colonne	Base de données NoSQL distribuée
Modèle de données	Fichiers bruts	Tables colonnaires	Clés-valeurs
Performance en lecture	Rapide pour les lectures séquentielles	Très rapide pour les agrégations et les requêtes analytiques	Performant pour les lectures et les écritures en parallèle
Performance en écriture	Rapide pour les écritures en bloc	Peut-être plus lent pour les mises à jour fréquentes	Performant pour les écritures en vrac

Exemples d'utilisation	Projets Hadoop, stockage de données brutes pour l'analytique	Analytique en temps réel, entrepôts de données, OLAP	Applications web à grande échelle, IoT, traçabilité des données
Licence	Open source	Open source	Open source

Tableau 7. Tableau comparatif des sources de données

### Annexe 3 : Courbe d'apprentissage du modèle

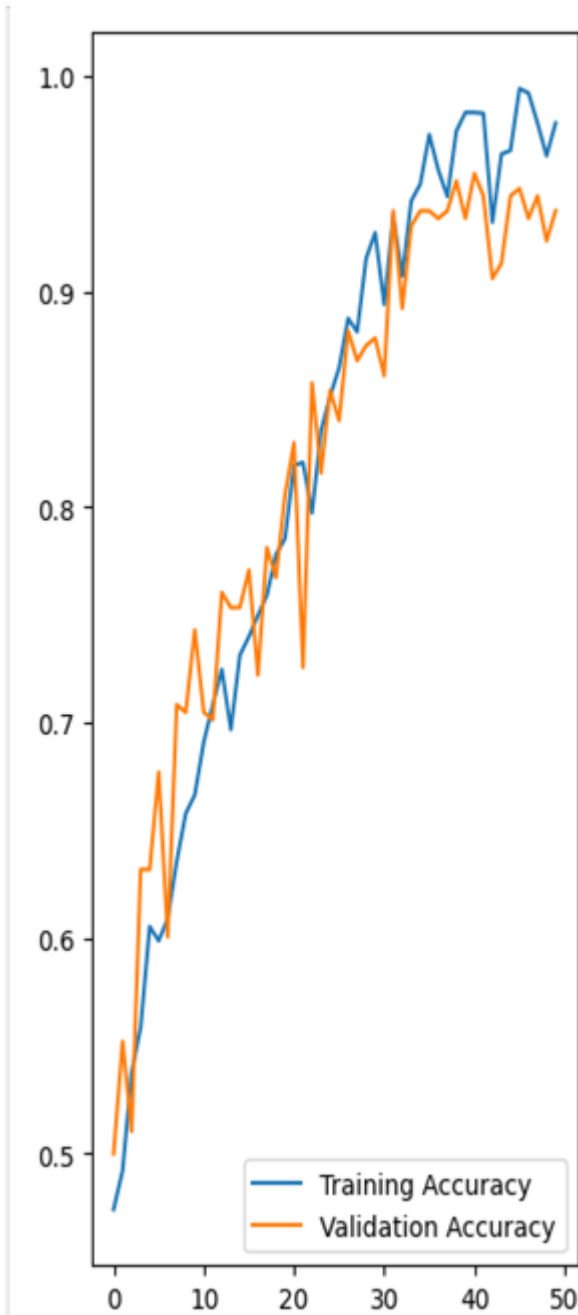


Figure 32. Courbe d'évolution de la précision du modèle

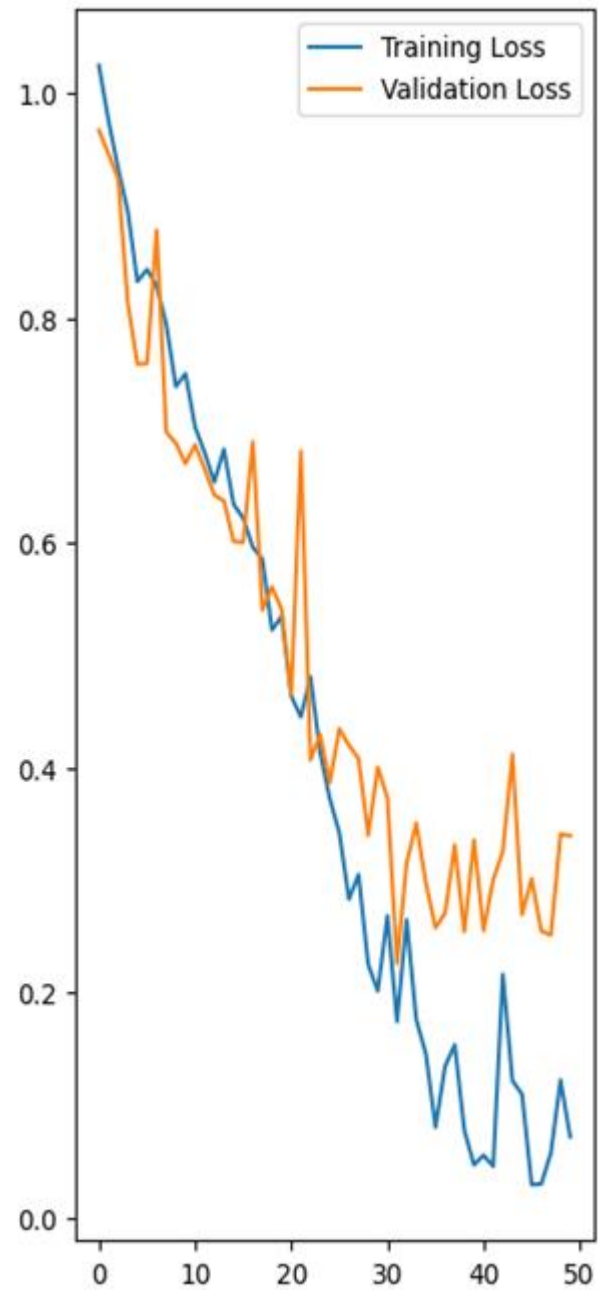


Figure 33. Courbe d'évolution du taux de perte du modèle

## 11.3 Annexe 4 : Comparaison de R et Python

Critères	R	Python
Popularité	Moins populaire dans l'IA, mais largement utilisé en statistiques	Très populaire dans l'IA, avec de nombreuses bibliothèques et communautés actives
Apprentissage	Facile pour les statisticiens et les analystes de données	Facile pour les développeurs en général, grande communauté d'apprentissage
Performance	Moins performant pour l'apprentissage profond en raison de limitations en termes de parallélisme	Performant pour l'apprentissage profond grâce à des bibliothèques optimisées
Support	Communauté plus petite, mais solide dans les statistiques	Grande communauté, support étendu, nombreuses ressources en ligne

Tableau 8. Comparaison entre R et Python