

Faculté des Sciences Semlalia de Marrakech (FSSM) Master Intelligent Artificielle

Système Agricole Intelligent



RÉALISÉ PAR :

Boujad Fatima-Zahra
Boumoula Abdelouafi
Amzil Asmaa
Amizmiz Habibatou Allah



ENCADRER PAR:

Dr. AMEKSA Mohammed



Membres du Jury:

Dr. AMEKSA Mohammed
Dr. Mjahed Soukaina
Dr. Chadi Mohamed-Amine

REMERCIEMENTS

Dans le cadre de ce projet , nous souhaitons tout d'abord exprimer notre gratitude envers Dieu qui nous a accordé la force, la patience et le courage nécessaires pour mener à bien ce travail.

Nous adressons nos remerciements les plus sincères à nos parents pour leur soutien inconditionnel, tant sur le plan moral qu'émotionnel, tout au long de notre parcours académique. Leur présence constante et leurs encouragements ont été une source de motivation précieuse pour surmonter chaque défi rencontré.

Nous tenons à exprimer notre profonde gratitude à Dr. AMEKSA Mohammed, notre encadrant, pour son soutien exceptionnel tout au long de ce projet. Grâce à son expertise, ses recommandations éclairées et sa disponibilité, il a su nous guider dans les moments difficiles, particulièrement lors de l'utilisation du matériel technique et des étapes complexes du développement du projet. Son engagement, sa patience et ses précieux conseils ont joué un rôle déterminant dans la réussite de ce travail.

Nous adressons également nos remerciements les plus sincères aux membres du jury :

Dr. AMEKSA Mohammed,

Dr. Mjahed Soukaina,

Dr. Chadi Mohamed-Amine,

pour avoir accepté d'évaluer notre travail. Leur présence, leurs observations pertinentes et leurs remarques constructives nous honorent et nous permettront d'améliorer encore davantage nos compétences.

Nous exprimons également notre reconnaissance envers les enseignants et le personnel administratif de la Faculté des Sciences Semlalia pour leur engagement sans relâche, leur passion pour l'enseignement et la qualité de la formation qu'ils nous offrent. Leur dévouement a largement contribué à l'acquisition des connaissances et compétences nécessaires pour mener à bien ce projet.

Nous remercions chaleureusement nos collègues de travail pour leur collaboration précieuse, leur esprit d'équipe et les échanges constructifs qui ont enrichi cette expérience. Leur implication a été un atout majeur dans l'aboutissement de ce projet.

Enfin, nous exprimons notre gratitude à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail, en nous apportant leur soutien, leurs encouragements et leurs conseils avisés.

Sommaire

Liste des figures	5
Liste des abréviations	7
Résumé	8
Abstract	9
Introduction générale	10
CHAPITRE I : Élaboration du Cahier des Charges	
INTRODUCTION.....	11
1- Contexte du projet	12
2- Problématique	12
3- Objectifs	13
4- Besoins fonctionnels	14
5- Besoins non fonctionnels	15
6- Description fonctionnelle et périmètre du projet	15
7- Planification du projet	15
7.1 L'équipe du projet	15
7.2 Les phases du projet	16
CONCLUSION.....	17
CHAPITRE II : Étude préalable du projet	
INTRODUCTION.....	18
1. État de l'Art :	19
1.1 Définitions :	19
Support Vector Machine	19
Naïve Bayes	19
Neural Network	19
Logistic Regression	19
K-Nearest Neighbors	19
1.2 Comparaison des Modèles de Classification pour la Détection de l'État des Pompes :	21-22
1.3 Méthodologie Adoptée :	22-27
1.3.1 Choix des Techniques d'Intelligence Artificielle :	22
1.3.2 Processus de Collecte de Données :	22-23
1.3.3 Protocole de Prétraitement :	23



1.3.4 Méthodes de Validation :	23
1.3.5 Outils et Technologies Utilisés :	23-26
1.4 Résultats et Discussion :	27-31
1.4.1 Contexte :	27
1.4.2 Collecte et Stockage des Données :	28-30
1.4.3 Ensemble de Données :	29
1.5 Analyse des Performances :	31
1.6 Stockage et Gestion des Données :	31
1.7 Limitations et Perspectives :	31
1.7.1 Limitations Actuelles :	31
1.7.2 Perspectives d'Évolution :	32
2.1 État de l'Art :	33
2.1.1 Bi-CNN :	33
2.1.2 ConvLSTM U-Net :	34
2.1.3 SVM avec Segmentation d'Image :	34-35
2.1.4 YOLOv5 :	35
2.1.5 Hybride CNN-SVM :	35-47
2.2 Comparaison des Modèles CNN pour la Classification des Maladies des Plantes :	37-39
2.3 Méthodologie Adoptée :	39
2.3.1 Choix des Techniques d'Intelligence Artificielle :	39
2.3.2 Choix et Caractéristiques du Dataset pour la Classification des Maladies :	40
2.3.3 Stockage et Gestion des Données Capturées :	40
2.3.4 Protocole de Prétraitement :	41
2.3.5 Méthodes de Validation :	41
2.3.6 Infrastructure de Développement :	41
2.3.7 Méthode d'Automatisation pour le Diagnostic des Plantes :	43
2.4 Flux de Traitement et Notifications :	44
2.4.1 Fonctionnement automatise.....	44-45
2.5 Résultats et Discussion :	45
2.5.1 Contexte et Implémentation :	45
2.5.2 Analyse des Performances :	45
2.5.3 Stockage et Gestion des Données :	46
2.5.4 Limitations Actuelles et Approches Futures :	46
2.5.5 Perspectives d'Évolution :	46-47
CONCLUSION:.....	48
CHAPITRE III : Question de recherche	
INTRODUCTION.....	49
1. Méthodes d'Apprentissage Automatique Étudiées	50
2. Architectures de Modèles	50



2-2. VGG16	51
2-3. Random Forest (HOG/LBP)	51
2-4. MobileNet (Embarqué)	52
3. Gestion des Conditions Variables	52
4. Comparaison des Performances	52
CONCLUSION	53
CHAPITRE IV : Intégration de LSTM	
INTRODUCTION	54
1. Apprentissage automatique	55
1.1 Réseau de neurones récurrents (RNN)	55
1.2 Long Short-Term Memory (LSTM)	56
2. Pourquoi utiliser LSTM dans ce projet ?	57
2.1 Prédiction des besoins en irrigation avec LSTM	57
2.1.1 Choix du modèle	57
2.1.2 Préparation des données	57
2.1.3 Entraînement et prédiction	57
2.1.4 Modélisation et entraînement du modèle	57
2.2 Stockage des prédictions dans MongoDB	58
2.3 Automatisation des notifications avec Node-RED	58
2.3.1 Récupération des données	58
2.3.2 Formatage et envoi des notifications	58
2.3.3 Objectif des notifications	59
CONCLUSION	60
CHAPITRE IV : phase de réalisation	
INTRODUCTION	61
CONCLUSION	64



Liste des Figures

• Figure 1 : DHT22	23
• Figure 2 : RainSensor	24
• Figure 3 : Capteur d'humidité du sol	24
• Figure 4 : MiniMoteur Pompe Eau DC	24
• Figure 5 : Modulerelaix à semi-conducteurs 1 canal 5VDC	24
• Figure 6 : WebcamIntegration	24
• Figure 7 : Breadboard	25
• Figure 8 : JumperWires M-M	25
• Figure 9 : USB Cable	25
• Figure 10 : Batterie Pile de 9V	25
• Figure 11 : Connecteur à clips pour pile 9V	25
• Figure 12 : LoPy 4.0 Pycom	26
• Figure 13 : Expansion Board	26
• Figure 14 : Système d'irrigation proposé	27
• Figure 15 : Système d'irrigation du tableau de bord aux services	28
• Figure 16 : Le flux Node-Red pour données	29
• Figure 17 : Système de supervision Node-red	30
• Figure 18 : Répartition des catégories selon température/humidité	30
• Figure 19 : Architecture des Flux Node-RED	45
• Figure 20 : Système agricole intelligent via WhatsApp	46
• Figure 28 : Architecture pour suivi des plantes	53
• Figure 29 : Système intelligent pour prévisions d'irrigation	55
• Figure 30 : Simulation du système d'irrigation intelligent	56



Liste des Abréviations

Abréviation	Description
AI	Artificial Intelligence
MCU	Microcontroller Unit
GPU	Graphics Processing Unit
KNN	K-Nearest Neighbors
ML	Machine learning
SVM	Support Vector Machine
MQTT	Message Queuing Telemetry Transport
MQL	MongoDB Query Language
IoT	Internet of Things
NN	Neural Network



RÉSUMÉ

Dans ce rapport, nous présentons un système intelligent combinant irrigation automatisée et diagnostique des maladies des plantes, conçu pour optimiser les ressources hydriques et améliorer la santé des cultures. Ce système repose sur des technologies IoT et des algorithmes d'apprentissage automatique, répondant ainsi aux défis croissants liés à la gestion durable de l'eau et à la protection des cultures face aux maladies.

Le projet consiste à développer un système basé sur des capteurs connectés pour collecter en temps réel des données environnementales (humidité du sol, température, pluie) et phytosanitaires (analyse des plantes). Grâce à des modèles avancés comme LSTM et CNN, le système prévoit les besoins en irrigation et détecte précocement les maladies, activant automatiquement la pompe à eau ou générant des recommandations phytosanitaires.

Une interface utilisateur interactive a également été développée, permettant de visualiser les données en temps réel et de recevoir des notifications automatiques concernant l'irrigation et l'état des cultures. Ce rapport décrit les étapes de conception, les technologies utilisées, les algorithmes implémentés et les résultats obtenus, démontrant comment notre solution contribue à une agriculture plus durable, précise et résiliente.



ABSTRACT

This report presents a smart system that combines automated irrigation and plant disease management to optimize water usage and improve crop health in agriculture. Leveraging IoT technologies and advanced machine learning algorithms, the system addresses challenges related to sustainable water management and timely disease detection.

The system collects real-time environmental data (soil moisture, temperature, rainfall) and plant health information through connected sensors and imaging devices. Using LSTM models, it predicts irrigation needs with high accuracy, while CNN models detect plant diseases early. Based on these analyses, the system automates water pump control and provides actionable recommendations for disease management.

An interactive user interface enables real-time data visualization and notifications about irrigation and crop conditions. This report outlines the design process, technologies employed, algorithms implemented, and the results achieved, demonstrating how the solution contributes to sustainable, precise, and resilient agriculture.



INTRODUCTION GÉNÉRALE

L'agriculture moderne est confrontée à des défis majeurs, tels que la raréfaction des ressources hydriques, les impacts du changement climatique et les pertes liées aux maladies des cultures. Ces enjeux nécessitent des solutions innovantes pour répondre à la demande croissante en production agricole tout en préservant les ressources naturelles. Les systèmes d'irrigation traditionnels, souvent inefficaces et manuels, entraînent un gaspillage important d'eau, tandis que la détection tardive des maladies des plantes compromet la productivité et augmente l'utilisation excessive de produits chimiques.

Dans ce contexte, ce projet propose de concevoir un système d'irrigation intelligent et automatisé intégrant des technologies avancées, telles que l'Internet des Objets (IoT) et des algorithmes d'apprentissage automatique. Ce système collecte en temps réel des données environnementales (humidité du sol, température, pluie) et phytosanitaires (analyse des plantes), et les analyse pour optimiser la gestion de l'irrigation et la détection précoce des maladies.

Pour prédire l'état de fonctionnement de la pompe en temps réel, un modèle SVM (Support Vector Machine) est utilisé, assurant une classification précise basée sur les données collectées. En complément, un modèle LSTM (Long Short-Term Memory) est utilisé pour prévoir les besoins en irrigation sur les prochains jours en se basant sur les résultats fournis par le SVM. Parallèlement, des modèles de vision par ordinateur, comme les réseaux convolutifs (CNN), détectent rapidement les signes de maladies sur les cultures, permettant des interventions ciblées. Le système offre une interface utilisateur intuitive pour surveiller les données en temps réel et prendre des décisions éclairées, tout en envoyant des notifications automatiques via des outils tels que WhatsApp.

Ce projet répond aux besoins croissants de l'agriculture de précision en combinant efficacité, durabilité et résilience face aux défis environnementaux et phytosanitaires. Il s'inscrit dans une vision globale d'une agriculture plus intelligente, permettant aux agriculteurs d'améliorer leur productivité tout en préservant les ressources naturelles pour les générations futures.

Ce rapport synthétise les différentes étapes du projet et se compose de 4 chapitres :

- Chapitre 1 : Élaboration du Cahier des Charges.
- Chapitre 2 : Étude préalable du projet.
- Chapitre 3 : Question de recherche.
- Chapitre 4 : Intégration de LSTM.

CHAPITRE I:

Élaboration du Cahier des Charges

INTRODUCTION

L'objectif de ce chapitre est de situer notre projet dans son contexte global en présentant son cahier des charges. Nous introduirons le contexte et la problématique liés à l'agriculture intelligente, préciserons les objectifs, les besoins fonctionnels et non fonctionnels, et décrirons les fonctionnalités principales ainsi que le périmètre du projet. Enfin, nous détaillerons les phases du projet et leur planification pour assurer une organisation efficace de sa réalisation.

1- Contexte du projet:

L'agriculture moderne fait face à des défis majeurs liés à l'efficacité de l'utilisation des ressources, en particulier l'eau, dans un contexte de changement climatique, de croissance démographique et de lutte contre les maladies des cultures. Les méthodes traditionnelles d'irrigation, souvent manuelles et non optimisées, entraînent un gaspillage des ressources hydriques, une baisse de la productivité et des coûts élevés pour les agriculteurs, tout en ne permettant pas une détection proactive des maladies.

Dans cette optique, le projet vise à concevoir et implémenter un système d'irrigation intelligent basé sur des technologies innovantes, notamment l'Internet des Objets (IoT) et l'apprentissage automatique. En combinant une collecte en temps réel des données environnementales (humidité du sol, température, précipitations) avec des données sur l'état des plantes (par exemple, via des images de feuilles et des capteurs spécialisés), ainsi que des algorithmes prédictifs, le système permet d'optimiser l'utilisation de l'eau tout en surveillant et en diagnostiquant les maladies des plantes.

Grâce à l'intégration de capteurs connectés (DHT22, capteurs d'humidité et de pluie) et de dispositifs d'imagerie associés à des modèles avancés d'apprentissage automatique comme SVM, LSTM (Long Short-Term Memory) et des réseaux convolutifs (CNN), le système garantit une analyse précise des données environnementales et phytosanitaires. Cela permet non seulement de prendre des décisions fiables concernant l'irrigation, mais aussi de détecter précocement les signes de maladies, en déclenchant des interventions ciblées pour protéger les cultures.

Ce projet s'inscrit dans une vision globale de l'agriculture de précision, visant à améliorer la durabilité, l'efficacité et la rentabilité des pratiques agricoles, tout en renforçant la résilience des cultures face aux maladies. Il fournit ainsi une solution complète pour optimiser la gestion de l'eau et réduire les pertes liées aux maladies des plantes, contribuant à une agriculture plus intelligente et durable.

2- Problématique:

L'agriculture, bien qu'essentielle, représente à elle seule 70 % de la consommation mondiale d'eau douce. Cependant, dans un contexte de raréfaction des ressources hydriques, elle fait face à des défis majeurs. Les systèmes d'irrigation traditionnels, souvent inefficaces et non adaptés aux conditions spécifiques du sol et de l'environnement, entraînent un gaspillage important de l'eau. De plus, les agriculteurs manquent d'outils performants pour anticiper les besoins en eau des cultures, qui varient selon les changements climatiques et les caractéristiques du sol.

Les solutions actuelles nécessitent généralement une intervention manuelle, augmentant ainsi les efforts, les coûts et le risque d'erreurs. Par ailleurs, les systèmes existants s'appuient souvent sur des données imprécises ou des algorithmes inadaptés, limitant leur capacité à fournir des recommandations optimales pour une irrigation efficace. À cela s'ajoute un problème d'accès : les technologies avancées, comme les systèmes connectés et les modèles d'apprentissage automatique, restent hors de portée pour de nombreux agriculteurs, en raison de leur coût élevé et de la complexité technique qu'elles impliquent.

Dans ce contexte, une modernisation des pratiques agricoles s'impose pour garantir une gestion durable et efficace des ressources en eau.

3- Objectifs:

Pour répondre aux problématiques actuelles, le projet propose une solution innovante combinant l'automatisation de l'irrigation et la gestion des maladies des plantes grâce à des technologies avancées. En s'appuyant sur des capteurs IoT, le système collecte en temps réel des données environnementales (humidité du sol, température, précipitations) et phytosanitaires (état des plantes via des images ou capteurs spécialisés). Ces données sont analysées à l'aide de modèles d'apprentissage automatique, tels que les réseaux LSTM (Long Short-Term Memory) pour prévoir les besoins en irrigation, et des réseaux convolutifs (CNN) pour détecter précocement les maladies des cultures.

Cette approche permet une gestion autonome et ciblée, optimisant à la fois la consommation d'eau et l'utilisation des traitements phytosanitaires. L'irrigation est déclenchée uniquement lorsque nécessaire, réduisant ainsi le gaspillage, tandis que les maladies des plantes sont identifiées rapidement, limitant les pertes et minimisant l'usage excessif de pesticides. Le système est conçu pour être économique, modulaire et facilement déployable dans différents environnements agricoles, ce qui le rend accessible même aux petites exploitations.

Une interface utilisateur intuitive, accessible sur ordinateur ou smartphone, offre aux agriculteurs des recommandations automatisées et des notifications en temps réel sur les besoins en irrigation et l'état sanitaire des cultures. Cela leur permet de prendre des décisions éclairées pour améliorer la productivité, la durabilité et la résilience de leurs exploitations face aux défis climatiques et phytosanitaires.

En intégrant l'Internet des Objets (IoT) et l'intelligence artificielle, ce projet répond aux besoins croissants de l'agriculture de précision, tout en contribuant à une gestion efficace des ressources naturelles, à la durabilité des pratiques agricoles et à la protection des cultures.

4- Besoins fonctionnels :

Le projet vise à concevoir un système intelligent combinant irrigation automatisée et gestion des maladies des plantes, basé sur des technologies IoT et des algorithmes d'apprentissage automatique, afin d'optimiser l'utilisation des ressources en eau et d'améliorer la santé des cultures dans un cadre agricole.

Le système collecte en temps réel des données environnementales grâce à des capteurs connectés : un capteur d'humidité du sol pour déterminer l'état hydrique, un capteur DHT22 pour mesurer la température et l'humidité de l'air, et un capteur de pluie pour détecter les précipitations. En parallèle, des capteurs ou des dispositifs d'imagerie capturent des données phytosanitaires sur l'état des plantes. Ces informations sont stockées de manière sécurisée dans une base de données MongoDB pour permettre une analyse approfondie.

L'analyse des données, à la fois historiques et en temps réel, est assurée par des modèles d'apprentissage automatique avancés. Un modèle LSTM (Long Short-Term Memory) prédit avec précision les besoins en irrigation pour les jours à venir, minimisant ainsi le gaspillage d'eau. Simultanément, des algorithmes de vision par ordinateur, comme les réseaux convolutifs (CNN), détectent précocement les maladies des plantes en analysant des images. Ces prédictions permettent d'activer ou de désactiver automatiquement la pompe à eau en fonction des seuils définis : la pompe s'active lorsque l'humidité du sol descend en dessous d'un seuil critique et se désactive en cas de précipitations. En cas de détection d'une maladie, des recommandations sont générées pour traiter rapidement les cultures.

Le système comprend une interface utilisateur interactive, accessible via un navigateur web sur ordinateur ou smartphone. Ce tableau de bord permet la visualisation en temps réel des données environnementales, des prévisions d'irrigation, et des alertes concernant les maladies des plantes. Il offre également la possibilité de basculer entre les modes manuel et automatique pour le contrôle de la pompe et la gestion des interventions phytosanitaires.

Des notifications automatiques sont envoyées via WhatsApp ou directement sur le tableau de bord, informant l'utilisateur de l'état de l'irrigation, des niveaux d'humidité, des précipitations, des prévisions d'irrigation, ainsi que des anomalies détectées sur les cultures. Ces messages clairs et concis facilitent une prise de décision rapide et éclairée.

Le système est conçu pour être modulaire et extensible, permettant l'ajout futur de nouveaux capteurs ou fonctionnalités, comme des caméras pour une surveillance visuelle détaillée. Il garantit une haute disponibilité 24/7 et une réponse quasi-instantanée aux changements environnementaux et phytosanitaires. Les données collectées sont sécurisées pour éviter toute perte ou modification non autorisée.

Enfin, le modèle d'apprentissage automatique sera mis à jour régulièrement pour améliorer sa performance et sa robustesse face aux nouvelles données. Cela assure une prédition fiable des besoins en irrigation et une détection précise des maladies, contribuant à une gestion durable des ressources et à la santé des cultures.

5- Besoins non fonctionnels :

Le système doit garantir une fiabilité optimale, assurant un fonctionnement ininterrompu 24/7, et offrir une précision élevée, avec des prédictions exactes pour l'irrigation et la gestion des maladies des plantes. Il doit être accessible sur plusieurs plateformes (ordinateur, smartphone) grâce à un tableau de bord intuitif et interactif. La sécurité des données collectées et stockées dans MongoDB doit être priorisée, tout en assurant une scalabilité pour intégrer de nouveaux capteurs ou fonctionnalités à l'avenir, notamment pour une surveillance plus avancée des cultures.

6- Description Fonctionnelle et Périmètre du Projet:

Le projet vise à développer un système automatisé combinant irrigation intelligente et gestion des maladies des plantes. Le système collecte en temps réel des données environnementales (humidité du sol, température, pluie, humidité de l'air) et phytosanitaires (via capteurs ou imagerie). Ces données sont analysées à l'aide de modèles d'apprentissage automatique, tels que LSTM pour prédire les besoins en irrigation et CNN pour détecter les maladies des plantes. La pompe d'irrigation est activée ou désactivée automatiquement via un module relais, et des recommandations sont générées pour traiter les maladies détectées. Un tableau de bord interactif permet la visualisation en temps réel, le contrôle manuel ou automatique, et l'envoi d'alertes via Node-RED ou WhatsApp pour informer l'utilisateur sur les conditions environnementales, l'irrigation et la santé des cultures.

7- Planification du projet :

7-1 L'équipe du projet :

La réussite d'un projet repose sur plusieurs éléments essentiels tels que l'équipe de travail, son organisation et les choix technologiques adoptés. Notre équipe est composée de trois étudiants en Master Intelligence Artificielle à la Faculté des Sciences Semlalia Marrakech. Précédemment, nous avons obtenu une Licence en SMI.

- **Boujad Fatima-Zahra**
- **Boumoula Abdelouafi**
- **Amzil Asmaa**
- **Amizmiz Habibatou Allah**

7-2 Les Phases du projet :

Phase d'analyse et de conception :

Identification des besoins fonctionnels et non fonctionnels.

Conception des architectures matérielle et logicielle.

Phase de collecte des données :

Installation des capteurs (humidité, température, pluie) et dispositifs d'imagerie.

Configuration des dispositifs IoT pour la collecte des données environnementales et phytosanitaires.

Stockage des données dans une base MongoDB.

Phase de développement :

Implémentation du modèle LSTM pour prédire les besoins en irrigation.

Développement du modèle CNN pour la détection des maladies des plantes.

Création du système de contrôle automatique de la pompe et des alertes.

Développement d'un tableau de bord interactif.

Phase de tests et validation :

Validation du fonctionnement en conditions réelles.

Évaluation des performances des modèles (LSTM et CNN).

Ajustements pour améliorer la précision et la robustesse du système.

Phase de déploiement :

Installation et mise en service sur site.

Formation des utilisateurs pour l'exploitation du système.

Phase de suivi et maintenance :

Surveillance continue du système.

Mise à jour régulière des modèles d'apprentissage automatique pour une meilleure précision.

Maintenance technique et ajout éventuel de nouvelles fonctionnalités.

FIN DU CHAPITRE I

CONCLUSION

Au cours de ce chapitre, nous avons présenté le contexte global du projet en élaborant un cahier des charges clair et structuré. Nous avons exposé la problématique liée à la gestion inefficace de l'irrigation agricole ainsi que les objectifs principaux à atteindre. De plus, nous avons décrit en détail les besoins fonctionnels et non fonctionnels, le périmètre du projet, ainsi que les différentes phases de réalisation pour assurer une mise en œuvre efficace de notre système d'irrigation intelligent.

CHAPITRE II:

Etude préalable du projet

INTRODUCTION

Ce chapitre explore les bases nécessaires pour la conception et la mise en œuvre du système d'irrigation intelligent. Il s'articule autour de l'étude des concepts essentiels, notamment l'Internet des Objets (IoT), les algorithmes d'apprentissage automatique comme SVM, et les modèles de vision par ordinateur utilisés pour la détection des maladies des plantes.

Nous examinerons également les solutions logicielles et matérielles choisies pour assurer une collecte, un traitement et une analyse efficaces des données environnementales et phytosanitaires. Cette étude préalable constitue le socle théorique et technique sur lequel repose l'ensemble du projet.

1- État de l'Art:

1-1 Définitions :

Support Vector Machine

La Machine à Vecteurs de Support (SVM) est un classifieur discriminatif défini formellement par un hyperplan de séparation. Ainsi, en utilisant des données d'entraînement étiquetées, les SVM peuvent produire un hyperplan qui catégorise les nouvelles données non étiquetées. Ensuite, un ensemble d'entraînement ($y_i; x_i$; $i=1,2,\dots,n$ où $x_i \in \mathbb{R}^n$) représente le vecteur d'entrée et $y_i \in \mathbb{R}^n$, représente l'élément cible. De nombreux types de SVM ont été développés pour s'adapter à différents types et complexes.

Naïve Bayes

L'algorithme Naïve Bayes est un algorithme d'apprentissage automatique probabiliste basé sur le théorème de Bayes développé par Thomas Bayes (1702–1761). Ce théorème peut être exprimé comme la probabilité qu'un événement A se produise étant donné que B est déjà survenu. En considérant $X = (x_1, \dots, x_n)$ comme les caractéristiques et Y comme la variable de classe. La spécificité de l'algorithme NB est qu'il suppose que les caractéristiques sont indépendantes et que la modification d'une caractéristique.

Neural Network

Le réseau de neurones peut effectuer un mappage arbitraire d'un espace vectoriel à un autre. Ces réseaux neuronaux peuvent utiliser des informations a priori inconnues cachées dans les données, mais ils ne peuvent pas les extraire. Dans le formalisme mathématique, l'apprentissage signifie ajuster les coefficients de pondération afin que certaines conditions soient remplies. Pour définir un réseau neuronal, nous introduisons d'abord le modèle linéaire défini comme suit :

$$p(x) = \frac{1}{1 + e^{-\frac{x-\mu}{s}}}$$

Logistic Regression

La régression logistique est un modèle utilisé lorsque la variable dépendante est catégorielle, soit (0 ou 1), (Vrai ou Faux) ou (Activé ou Désactivé). La fonction logistique prend la forme suivante :

K-Nearest Neighbors

L'algorithme des k plus proches voisins (KNN) est un algorithme simple qui stocke tous les cas disponibles et classe les nouveaux cas en fonction d'une mesure de similarité. Si nous voulons classer un cas à l'aide de l'approche KNN, le cas est classé par un vote majoritaire de ses voisins, et le cas est attribué à la classe la plus proche parmi ses k voisins mesurée par une fonction de distance : ✓ Distance Euclidienne :

$$d(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

✓ Distance Manhattan :

$$d(x, y) = \sum_{i=1}^k |x_i - y_i|$$

✓ Distance Minkows :

$$d(x, y) = \left(\sum_{i=1}^k |x_i - y_i|^q \right)^{\frac{1}{q}}$$

Tableau1.Travaux antérieurs sur les modèles d'apprentissage supervisé.

Référence	Modèle supervisé	Caractéristiques	Expérimental	Simulation
[17]	Régression linéaire	Prédiction de l'eau d'irrigation requise via des capteurs	✓	✓
[18]	KNN, SVM, Régression logistique	Système de prédiction en temps réel de l'eau pour l'irrigation	✓	✗
[19]	SVM, KNN, Naïve Bayes	Classification basée sur un seuil dans une base de données cloud.	✓	✓
[20]	KNN, SVM	Détection d'infections avec surveillance en temps réel de l'humidité et de la température.	✓	✓
[21]	KNN	Analyse et surveillance d'images agricoles prises par des drones.	✓	✓
[22]	SVM	Ajustement automatique de l'irrigation dans un environnement domestique.	✓	✓
[23]	KNN, Réseaux de neurones, Naïve Bayes, SVM, Régression logistique	Approche d'irrigation intelligente utilisant plusieurs algorithmes pour la prédiction. Précision maximale atteinte : 98,3%	✓	✓
		pour le modèle KNN.		

1-2 Comparaison des Modèles de Classification pour la Détection de l'État des Pompes :

Dans le cadre de ce projet, plusieurs modèles d'apprentissage automatique ont été entraînés et évalués pour détecter l'état de fonctionnement des pompes en utilisant un jeu de données recueilli à partir de capteurs industriels. L'objectif était de comparer leurs performances en utilisant les métriques suivantes : précision (accuracy), F1-score, précision (precision), et rappel (recall), afin d'identifier le modèle le plus performant pour ce contexte spécifique. Les modèles étudiés incluent KNN, réseau de neurones (Neural Network), machine à vecteurs de support (Support Vector Machine), régression logistique (Logistic Regression), et Naïve Bayes, chacun offrant une approche complémentaire pour la classification.

Modele	Accuracy	F1 Score	Precision	Recall
K-Nearest Neighbors	92.5%	0.90	0.91	0.89
Neural Network	99%	0.98	0.99	0.97
Support Vector Machine	99.95%	0.99	0.99	0.99
Logistic Regression	99.85%	0.99	0.99	0.99
Naive Bayes	92.5%	0.91	0.92	0.90

Interprétation des Résultats :

Support Vector Machine se distingue par la meilleure performance globale avec une accuracy de 99.95% et des scores de F1, précision et rappel atteignant 0.99, ce qui en fait le modèle le plus fiable pour ce problème.

Logistic Regression présente des performances très proches, avec une accuracy de 99.85% et des métriques similaires, en offrant une approche plus simple et rapide à entraîner.

Neural Network atteint une accuracy de 99% avec un très bon compromis entre précision et rappel, en confirmant sa capacité à gérer des relations complexes dans les données.

K-Nearest Neighbors (KNN) et Naïve Bayes montrent des performances plus faibles avec une accuracy de 92.5%, bien que le F1 score et le rappel indiquent qu'ils restent efficaces pour des tâches moins exigeantes.

Naïve Bayes surpassé légèrement KNN avec des scores légèrement meilleurs en F1 score et rappel, mais reste limité face aux modèles plus avancés.

Ces résultats soulignent que les modèles SVM et Logistic Regression sont les plus adaptés pour cette tâche, alliant précision et robustesse.

1-3 Méthodologie Adoptée:

1-3-1 Choix des Techniques d'Intelligence Artificielle

La sélection de l'algorithme d'apprentissage automatique a suivi un processus méthodique visant à identifier la technique la plus adaptée à notre problématique de décision d'irrigation. Après une analyse comparative approfondie, les Machines à Vecteurs de Support (SVM) ont été retenues pour leurs capacités remarquables en classification. Le choix des SVM repose sur plusieurs critères fondamentaux : leur aptitude à résoudre des problèmes de classification non linéaires, leur robustesse face à des données de dimensions élevées et leur efficacité dans la généralisation. Contrairement à d'autres algorithmes tels que la régression logistique ou les arbres de décision, les SVM se distinguent par leur capacité à créer des hyperplans de décision optimaux, permettant une séparation précise des différentes classes de décisions d'irrigation. Un noyau linéaire a été sélectionné pour sa simplicité computationnelle et sa capacité à traiter efficacement nos données environnementales. Cette approche optimise la séparation des classes en transformant l'espace des caractéristiques de manière adéquate.

1-3-2 Processus de Collecte de Données

Caractéristiques du Jeu de Données

Le jeu de données a été conçu pour capturer de manière exhaustive les paramètres environnementaux essentiels à la prise de décision en matière d'irrigation. Il comprend les caractéristiques suivantes :

Capteur de Pluie : Donnée binaire (1 pour pluie, 0 pour absence de pluie).

Plage: {0, 1}

Objectif : Identifier la présence ou l'absence de précipitations.

Humidité du Sol : Valeurs exprimées en pourcentage.

Plage: 10-100 %

Résolution : Précision à 1 %

Objectif : Évaluer l'état hydrique du sol.

Température : Mesurée en degrés Celsius.

Plage: 5-40 °C

Résolution : Précision au dixième de degré

Objectif : Caractériser les conditions thermiques ambiantes.

Humidité de l'Air : Valeurs exprimées en pourcentage.

Plage: 20-90 %

Résolution : Précision à 1 %

Objectif : Mesurer l'humidité atmosphérique.

Statut de la Pompe : Donnée binaire (1 pour activée, 0 pour désactivée).

Plage: {0, 1}

Objectif : Enregistrer l'état opérationnel du système d'irrigation.

Volume et Répartition des Données

Le jeu de données contient 10 000 échantillons répartis comme suit :

Ensemble d'entraînement : 8 000 échantillons (80 %)

Ensemble de test : 2 000 échantillons (20 %)

Cette répartition garantit une validation robuste du modèle, avec un équilibre entre données disponibles pour l'entraînement et un ensemble de tests significatif pour l'évaluation.

Stockage et Gestion des Données

Chaque échantillon est sauvegardé dans une base de données MongoDB locale, selon une structure documentaire standardisée. Les étapes principales du processus de stockage incluent :

1. Collecte des Données : Transmission des mesures via le microcontrôleur LoPy 4.0.
2. Prétraitement : Normalisation et vérification de la cohérence des données.
3. Horodatage : Ajout d'un horodatage précis à chaque enregistrement.
4. Stockage : Insertion dans MongoDB avec sauvegarde et réplication dans un stockage cloud secondaire.

Ce processus garantit une traçabilité complète, facilite les requêtes historiques, et sert de base pour l'apprentissage continu du modèle

1-3-3 Protocole de Prétraitement

Le prétraitement des données constitue une étape cruciale pour assurer la qualité et la fiabilité du modèle. Les techniques appliquées incluent :

- Normalisation : Transformation des caractéristiques pour une moyenne nulle et un écart-type unitaire, afin de garantir une contribution équitable des capteurs au modèle.
- Filtrage des Valeurs Aberrantes : Suppression des mesures incohérentes sur la base de seuils définis dynamiquement selon les distributions statistiques.
- Imputation des Données Manquantes : Utilisation de l'interpolation linéaire pour combler les valeurs manquantes sans introduire d'artéfacts.

1-3-4 Méthodes de Validation

La validation du modèle SVM a été réalisée à l'aide d'une approche rigoureuse :

- Validation croisée K-fold ($k=10$) : Division du jeu de données en sous-ensembles, où chaque sous-ensemble est utilisé à tour de rôle pour le test et l'entraînement.
- Métriques d'évaluation :

Précision globale.

Matrice de confusion.

ScoreF1.

Erreur quadratique moyenne (RMSE).

Des techniques de régularisation ont été employées pour éviter le surapprentissage, et les courbes d'apprentissage ont été analysées pour surveiller le comportement du modèle.

1-3-5 Outils et Technologies Utilisés

L'infrastructure technologique du projet repose sur une combinaison d'outils et de matériels avancés pour répondre aux besoins spécifiques du projet : Microcontrôleur et Matériel IoT Expansion Board : Carte d'expansion utilisée pour simplifier l'intégration et l'alimentation des modules connectés.

- DHT22: Capteur numérique de température et d'humidité, utilisé pour surveiller les conditions climatiques et optimiser les paramètres agricoles.



Figure 1: DHT22

- RainSensor : Capteur permettant de détecter la présence ou l'absence d'eau, essentiel pour automatiser l'irrigation en fonction des précipitations.



Figure 2: RainSensor

- Capteur d'humidité du sol : Instrument clé pour mesurer l'humidité du sol, garantissant un arrosage précis et adapté.



Figure 3: Capteur d'humidité du sol

- MiniMoteur Pompe Eau DC: Pompe compacte permettant d'automatiser l'irrigation dans le système.



Figure 4: MiniMoteur Pompe Eau DC

- Modulerelais à semi-conducteurs 1 canal 5VDC : Relais conçu pour contrôler des dispositifs à haute puissance comme les pompes ou les lumières.



Figure 5: Modulerelais à semi-conducteurs 1 canal 5VDC

- WebcamIntegration : Webcam utilisée pour capturer des images ou des vidéos afin de surveiller visuellement les cultures. Prototypage et Connexions



Figure 6: WebcamIntegration

- Breadboard : Planche de prototypage permettant une configuration rapide et sans soudure des circuits électroniques.

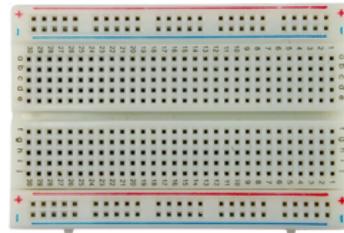


Figure 7: Breadboard

- JumperWires M-M: Fils de connexion mâle-mâle pour relier les composants entre eux.



Figure 8: JumperWires M-M

- USBCable: Utilisé pour alimenter les dispositifs électroniques et transférer des données.



Figure 9: USBCable

Alimentation

- Batterie Pile de 9V : Une source d'énergie compacte et portable pour les modules électroniques.



Figure 10: Batterie Pile de 9V

- Connecteur à clips pour pile 9V : Permet de connecter rapidement et facilement une pile 9V au circuit



Figure 11: Connecteur à clips pour pile 9V

- LoPy 4.0 Pycom : Microcontrôleur avec connectivité LoRa, Wi-Fi et Bluetooth, idéal pour des projets IoT agricoles.

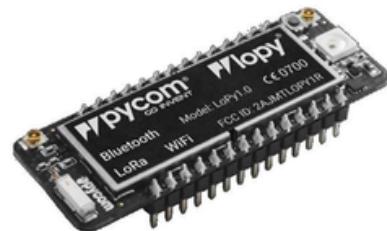


Figure 12: LoPy 4.0 Pycom

- Expansion Board

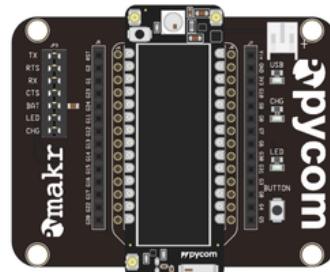


Figure 13: Expansion Board

Infrastructure de Développement

- Langage de programmation : Python, avec des bibliothèques comme scikit-learn pour les algorithmes d'apprentissage automatique, pandas pour la gestion des données et NumPy pour les calculs numériques.
- Environnement de développement : Jupyter Notebook, facilitant le développement interactif et l'analyse des données.
- Stockage des données : MongoDB, utilisée pour le stockage des données, avec des options d'implémentation locale et cloud.

Cette combinaison d'outils et de technologies forme une base robuste pour le développement d'un système agricole intelligent, assurant à la fois fiabilité, flexibilité et efficacité

1-4 Résultats et Discussion:

1-4-1. Contexte

Ce rapport présente les différentes étapes suivies pour concevoir et mettre en œuvre un système d'irrigation intelligent. La figure 1 illustre l'ensemble des étapes nécessaires à la réalisation de ce système. La première étape a consisté à sélectionner les capteurs adaptés au projet, notamment :

- Uncapteur d'humidité du sol pour mesurer le niveau d'humidité du sol.
- Descapteurs de température et d'humidité de l'air.
- Uncapteur de pluie

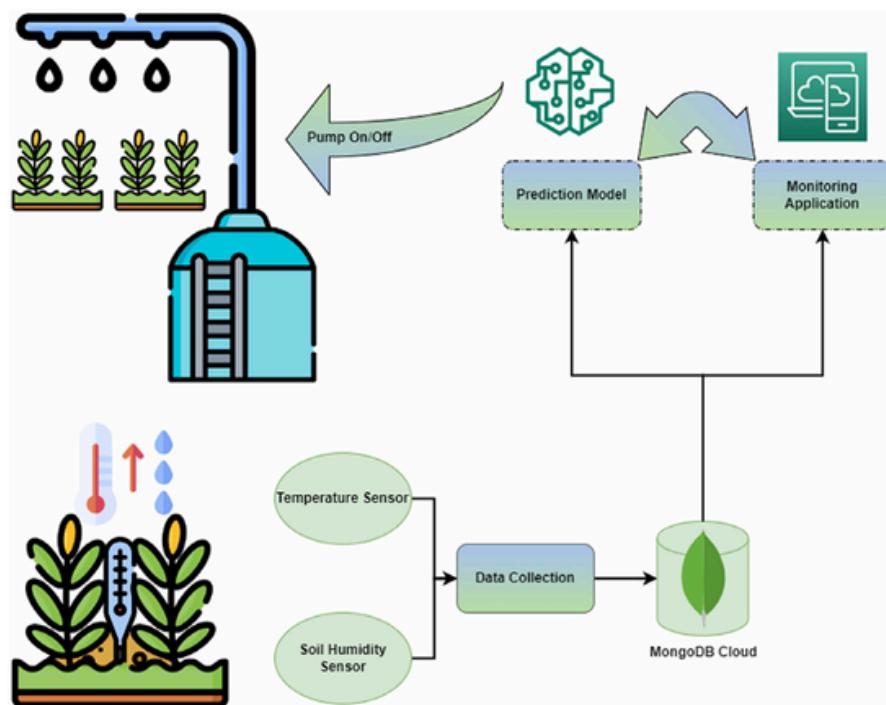


Figure 14: Système d'irrigation proposé.

Une fois les capteurs connectés à la carte LoPy4, nous avons programmé cette dernière afin d'assurer la collecte et la transmission en temps réel des données. Ces données sont ensuite stockées à l'aide de la plateforme Node-RED et de MongoDB pour permettre l'analyse et la prise de décisions automatiques, telles que l'activation ou la désactivation de la pompe d'irrigation.

1-4-2. Collecte et stockage des données.

La plateforme Node-RED a été utilisée pour connecter le système à des outils de supervision et de stockage, tout en permettant l'envoi de notifications via WhatsApp. Cette étape a impliqué la récupération, la visualisation et le prétraitement des données issues des capteurs (voir Fig. 2)

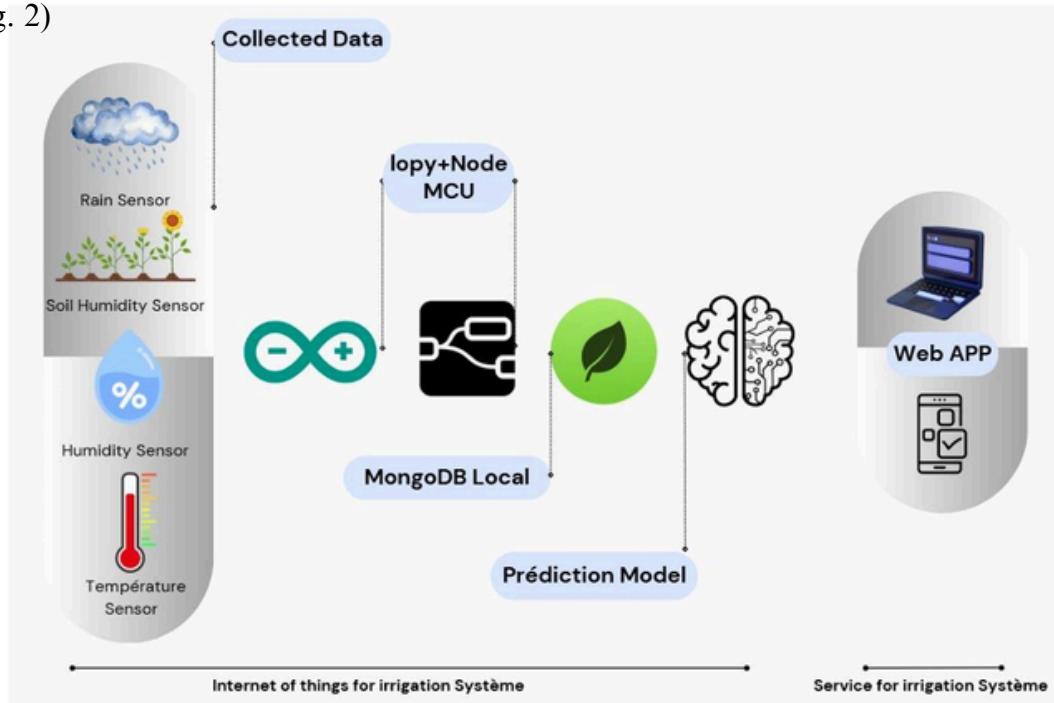


Figure 15: Système d'irrigation du tableau de bord aux services.

Les données collectées peuvent être regroupées en deux catégories principales :

1. Données issues des capteurs :

Température.

Humidité de l'air.

Humidité du sol.

Précipitations.

2. Données opérationnelles :

États de fonctionnement de la pompe (marche/arrêt).

La figure 16 illustre le flux de données utilisé pour leur acquisition et leur stockage, composé des éléments suivants :

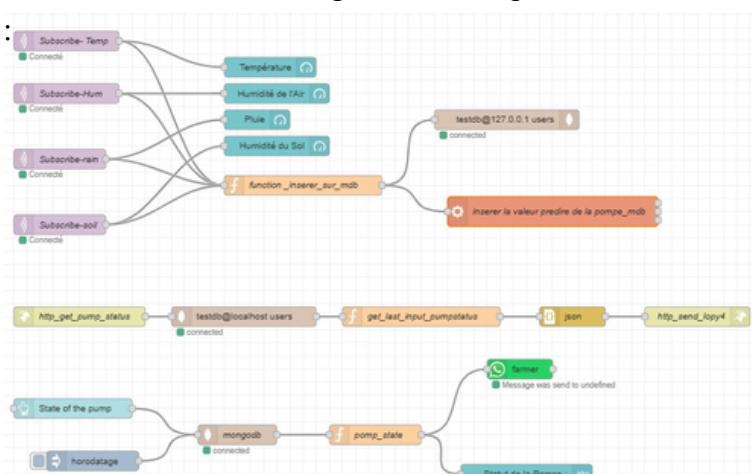


Figure 16: Le flux Node-Red est utilisé pour diviser, stocker, notifier et visualiser les données..

- NœudsMQTT: Permettent la connexion entre le serveur Node-RED et la carte LoPy4.
- Nœuddeprétraitement : Segmentent et structure les données collectées.
- Nœudsspécifiques : Gèrent les données de chaque capteur (humidité du sol, température, humidité de l'air, précipitations).
- Nœudsdenotification WhatsApp : Assurent un suivi en temps réel en envoyant des alertes pertinentes.
- NœudsMongoDB: Permettent le stockage des données dans une base de données MongoDB.
- NœudsDashboard: Offrent une visualisation en temps réel des données collectées (voir Fig. 17)

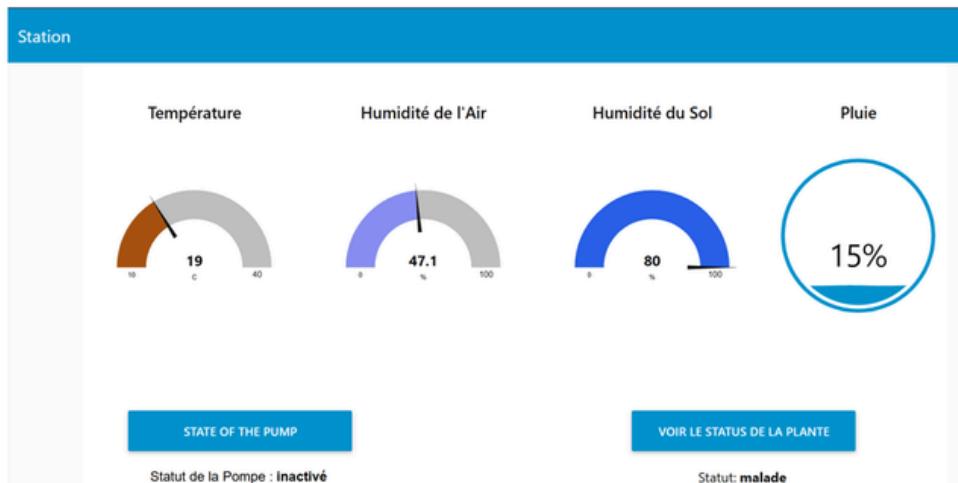


Figure 17: Système de supervision Node-red.

Le stockage des données repose sur MongoDB, configuré en mode local via une connexion Node-RED. Cette méthode permet d'entraîner les modèles prédictifs utilisés pour optimiser le fonctionnement du système.

1-4-3. Ensemble de données.

Grâce à l'intégration des technologies IoT, des dispositifs autonomes, tels que des capteurs connectés, ont été déployés dans divers environnements.

Ces dispositifs ont permis une collecte de données massives, en particulier pour des plantes domestiques, dans le but de répondre aux besoins en information pour l'irrigation intelligente.

Les données collectées comprennent :

1. Données d'humidité du sol :

Plage: [314,47- 987,83].

Moyenne:384,5. 2.

2.Données de température :

Plage: [18 °C- 39 °C].

Moyenne:26,34 °C. 3.

3.Données d'humidité de l'air :

Plage: [38 %- 81,3 %].

Moyenne:66,4 %.

Les données issues des capteurs ont été analysées et utilisées pour générer des sorties catégoriques afin de contrôler la pompe :

- 0: Indique que la pompe doit être arrêtée.
- 1: Indique que la pompe doit être activée.

Cette architecture repose sur un principe pair-à-pair (peer-to-peer), favorisant une gestion décentralisée et évolutive.

L'implémentation actuelle démontre la faisabilité du système tout en posant les bases d'une extension future.

Ce graphique, illustré à la figure 18, propose une manipulation schématique des données visant à modéliser de manière réfléchie les résultats obtenus. Cette approche permet de rendre compte du plan de dentition en le concrétisant comme un résultat spécifique. Dans ce cadre, les prévisions restent largement inductives par nature. Par la suite, des observations empiriques sont résumées de manière universelle grâce à une identification entre pairs, qui distingue deux catégories représentées par des couleurs différentes.

La première catégorie, associée à la couleur rouge, symbolise la connaissance de niveau "0". Elle est déterminée par un pompage dépendant de la température, avec une humidité orientée vers la désactivation. La seconde catégorie, représentée par la couleur verte, correspond au niveau "1". Elle est basée sur un pompage influencé par la température, mais également par une humidité active.



Figure 18: Répartition des catégories selon la température et l'humidité.

La présentation des sections prédéfinies des plantes repose sur l'utilisation de données massives, combinées à un entraînement à l'aide d'algorithmes SVM (Support Vector Machines). Cette approche permet d'extrapoler les informations des événements passés pour mieux prévoir les événements futurs et établir des projections pour notre système d'irrigation. L'objectif est de promouvoir les tendances futures susceptibles de se concrétiser.

1-5 Analyse des Performances:

Les résultats de notre système d'irrigation intelligent démontrent une capacité efficace de collecte et de stockage des données. L'analyse des données collectées dans notre base MongoDB locale révèle une bonne précision dans la détection des conditions environnementales et la prise de décision pour l'irrigation. Notre système combine deux approches d'apprentissage automatique :

Le système SVM implémenté a démontré une précision de 99,95% dans la classification des états nécessitant une irrigation, tandis que l'utilisation des réseaux LSTM a permis d'améliorer la prédiction temporelle des besoins en irrigation en prenant en compte les séquences historiques des données.

1-6 Stockage et Gestion des Données:

Notre système utilise une base de données MongoDB locale comme solution de stockage principale. Cette approche nous permet de stocker efficacement les données des capteurs et de les organiser de manière structurée. Les données stockées comprennent les mesures d'humidité du sol (plage de 314,47 à 987,83), de température (18°C à 39°C), d'humidité de l'air (38% à 81,3%), ainsi que l'état de la pompe (activée ou désactivée).

1-7 Limitations et Perspectives:

1-7-1 Limitations Actuelles.

Bien que notre implémentation actuelle avec SVM et LSTM offre de bonnes performances, plusieurs axes d'amélioration ont été identifiés :

La calibration dynamique des capteurs pourrait être implémentée pour permettre un ajustement automatique des seuils en fonction des conditions saisonnières. Un système de filtrage adaptatif pourrait être ajouté pour réduire le bruit des capteurs et améliorer la précision des mesures.

Une gestion énergétique plus sophistiquée pourrait être mise en place, incluant un mode veille intelligent durant les périodes d'inactivité et une optimisation des cycles de transmission des données. Ces améliorations permettraient de réduire significativement la consommation énergétique du système.

Le traitement des données pourrait être enrichi par l'ajout de mécanismes de compression avant transmission et de mise en cache locale pour améliorer la résilience aux pannes réseau.

1.7.2 Perspectives d'Évolution

Pour les développements futurs, plusieurs pistes d'amélioration sont envisageables :

L'architecture actuelle combinant SVM et LSTM pourrait être enrichie par l'intégration d'autres algorithmes d'apprentissage automatique comme les forêts aléatoires ou les réseaux de neurones convolutifs pour la détection d'anomalies dans les données des capteurs.

La scalabilité du système pourrait être améliorée par une migration vers une architecture distribuée, permettant la gestion de plusieurs zones d'irrigation simultanément. L'ajout d'un système de réPLICATION DES DONNÉES améliorerait la robustesse et la fiabilité du système.

L'interface utilisateur pourrait être enrichie par le développement d'une application mobile offrant une visualisation en temps réel des données et un contrôle plus précis du système d'irrigation. Cette interface pourrait également afficher les prédictions générées par nos modèles SVM et LSTM pour une meilleure anticipation des besoins en irrigation.

La sécurité pourrait être renforcée par l'implémentation de mécanismes de chiffrement des données et de détection d'anomalies.

Cette approche progressive d'amélioration, basée sur notre implémentation actuelle combinant MongoDB, SVM et LSTM, permettrait d'évoluer vers une solution plus complète et robuste pour l'agriculture intelligente.

2-1 État de l'Art:

2.1 Définitions :

Nous présentons une analyse comparative de plusieurs modèles d'intelligence artificielle utilisés pour la classification des maladies des plantes à partir d'images de feuilles. Chaque modèle utilise des techniques d'apprentissage automatique spécifiques qui présentent des avantages et des inconvénients selon leur complexité, leur précision et leurs exigences en termes de données. Nous avons examiné les modèles Bi-CNN, ConvLSTM U-Net, SVM avec segmentation d'image, YOLOv5 et un modèle hybride CNN-SVM.

2.1.1 Bi-CNN (Bilinear Convolutional Neural Networks)

Le modèle Bi-CNN utilise un réseau de neurones convolutifs bilinéaires pour extraire et combiner des caractéristiques fines des images de feuilles. La principale innovation de ce modèle réside dans l'application du pooling bilinéaire, qui permet de combiner les caractéristiques extraites de deux réseaux convolutifs différents pour former un vecteur plus riche. Mathématiques associées :

Extraction des caractéristiques : Les caractéristiques sont extraites par deux réseaux convolutifs, chacun produisant des matrices de caractéristiques F_x et F_y :

$$F_X = [f_{x1}, f_{x2}, \dots, f_{xm}], F_Y = [f_{y1}, f_{y2}, \dots, f_{yn}]$$

Pooling bilinéaire : Les matrices sont combinées par multiplication pour former un vecteur bilinéaire B :

$$B = F_X^T F_Y$$

Normalisation :

Une normalisation L2 est appliquée pour améliorer la stabilité du modèle :

$$\hat{B} = \frac{\sqrt{B}}{\|B\|}$$

Avantages et Inconvénients :

Avantages : Précision élevée (94,98%), automatisation de la détection, scalabilité et extraction fine de caractéristiques. ☒

Inconvénients : Complexité du modèle, dépendance aux données et besoin de modèles pré-entraînés.

2.1.2 ConvLSTM U-Net:

Le modèle ConvLSTM U-Net combine des réseaux de neurones convolutifs (CNN) et des réseaux de neurones récurrents à convolution (ConvLSTM). Cette combinaison permet de capturer à la fois les caractéristiques spatiales et temporelles des images de feuilles. Le modèle utilise une architecture en U-Net pour la segmentation des images et un réseau ConvLSTM pour traiter les informations temporelles.

Mathématiques associées

- Convolution classique :
- La fonction de convolution est appliquée pour extraire les caractéristiques spatiales de l'image :

$$h_{ij} = \sum_m \sum_n I[i+m, j+n] \cdot K[m, n] + b$$

LSTM :

- Les portes LSTM capturent la dynamique temporelle :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

où f_t , i_t , o_t sont respectivement les portes d'oubli, d'entrée et de sortie, et c_t l'état mémoire.

Avantages et Inconvénients

Avantages :

Très haute précision (98,79 %).

Efficacité avec moins de paramètres.

Extensibilité à d'autres maladies.

Inconvénients :

Complexité d'implémentation.

Forte dépendance aux données.

2-1-3- SVM avec Segmentation d'Image:

Le modèle SVM avec segmentation d'image utilise des techniques classiques de traitement d'image et des classificateurs SVM pour identifier et classifier les maladies. Le prétraitement des images inclut des étapes de filtrage pour réduire le bruit, suivi de la segmentation avec des techniques comme Otsu ou le clustering k-means.

Mathématiques associées

Fonction de décision du SVM :

Le SVM cherche à séparer les classes avec un hyperplan donné par la fonction :

$$f(x) = \text{sign}(\langle w, x \rangle + b)$$

Optimisation du SVM :

L'objectif est de maximiser la marge :

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{sous contrainte : } y_i(\langle w, x_i \rangle + b) \geq 1, \forall i$$

Avantages et Inconvénients :

Avantages : Haute précision, flexibilité pour différentes cultures, réduction du besoin d'analyse manuelle. ☒ Inconvénients : Ressources computationnelles importantes, dépendance à la qualité des images et nécessité de données étiquetées.

2.1.4. YOLOv5

Le modèle YOLOv5 est une solution de détection d'objets en temps réel. Il est optimisé pour la vitesse et la légèreté tout en offrant une bonne précision pour la détection des maladies des plantes. Son architecture permet une détection rapide des objets en une seule passe. Mathématiques associées :

Fonction de perte : La détection utilise une combinaison d'erreurs de localisation, de classification et de confiance

$$\begin{aligned} \mathcal{L}_{\text{loc}} &= \sum_{i=1}^N \mathbf{1}_{\{\text{obj}\}}^i (\|x_i - \hat{x}_i\|^2 + \|y_i - \hat{y}_i\|^2) \\ \mathcal{L}_{\text{cls}} &= - \sum_{i=1}^N \mathbf{1}_{\{\text{obj}\}}^i \sum_{c=1}^C p_{i,c} \log(\hat{p}_{i,c}) \end{aligned}$$

$$\mathcal{L} = \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{conf}}$$

Avantages et Inconvénients :

Avantages : Détection rapide et précise, léger et adaptable à divers types de maladies. ☒

Inconvénients : Performance dépendante de la qualité des données et difficultés dans des environnements complexes.

2.1.5. Hybride CNN-SVM

Le modèle Hybride CNN-SVM combine les avantages des réseaux convolutifs (CNN) pour l'extraction des caractéristiques et des machines à vecteurs de support (SVM) pour la classification des maladies. Ce modèle est particulièrement adapté pour les tâches de classification où une précision élevée est requise.

Mathématiques associées :

Extraction par CNN : Les caractéristiques sont extraites par des convolutions successives :

$$f_t = \text{ReLU}(W_t \cdot f_{t-1} + b_t)$$

Classification par SVM : Après extraction des caractéristiques, un SVM est utilisé pour classifier les données :

$$f(x) = \text{sign}(\langle w, x \rangle + b)$$

Avantages et Inconvénients :

Avantages : Précision élevée (99%), flexibilité et robustesse dans la classification.

Inconvénients : Demande de grandes ressources computationnelles et de bonnes données.

Tableau Récapitulatif

Modèle	Avantages	Inconvénients	Performance
Bi-CNN	<ul style="list-style-type: none"> - Automatique - Précis (94,98%) - Flexible - Bonne extraction des caractéristiques 	<ul style="list-style-type: none"> - Complexes - Dépendant des données - Besoin de modèles pré-entraînés 	Précision : 94,98%
ConvLSTM U-Net	<ul style="list-style-type: none"> - Très précis (98,79%) - Efficace et rapide - S'adapte à d'autres maladies 	<ul style="list-style-type: none"> - Difficile à mettre en œuvre - Dépend de données variées 	Précision : 98,79%
SVM	<ul style="list-style-type: none"> - Précis - Flexible - Réduit le travail manuel 	<ul style="list-style-type: none"> - Dépend des images de qualité - Nécessite des données bien étiquetées 	Précision: 71%-96% selon les études

YOLOv5	<ul style="list-style-type: none"> - Rapide et efficace en temps réel - Léger - Adaptable 	<ul style="list-style-type: none"> - Limité dans des environnements complexes - Sensible aux données 	Précision : 92,57%
Hybride CNN-SVM	<ul style="list-style-type: none"> - Très précis (99%) - Combinaison puissante - Propose des solutions 	<ul style="list-style-type: none"> -Demande beaucoup de ressources 	Précision : 99%, Temps de traitement : 19-148sec
	de traitement	-Long à exécuter	

Cet étude présente les modèles de classification des maladies des plantes en détaillant leurs principes mathématiques et leurs performances dans différents contextes. Les résultats montrent que les modèles CNN, qu'ils soient hybrides ou basés sur Bi-CNN, sont particulièrement performants, mais exigent une infrastructure et des données de haute qualité pour optimiser leurs performances.

2.2 Comparaison des Modèles CNN pour la Classification des Maladies des Plantes :

Dans le cadre de ce projet, plusieurs modèles d'apprentissage automatique ont été entraînés et évalués pour la classification des maladies des plantes en utilisant le dataset New Plant Diseases Dataset. L'objectif était de comparer leurs performances en termes de précision, de perte, et de temps d'entraînement, afin d'identifier le modèle le plus adapté aux besoins pratiques et computationnels. Les modèles sélectionnés incluent MobileNetV2, Bi-CNN, ConvLSTM U-Net, YOLOv5, et un hybride CNN-SVM, chacun représentant une approche différente et complémentaire. Les résultats obtenus sont présentés dans le tableau ci-dessous, avec une description du processus suivi pour chaque modèle.

Modèle	Précision (%)	Perte	Temps d'entraînement (minutes)	Description du Processus
MobileNet V2	95.2%	0.15	30	Basé sur MobileNetV2 pré-entraîné (ImageNet), avec couches GlobalAveragePooling2D et Dropout (0.3). Fonction de perte : categorical_crossentropy. Optimiseur : Adam.
Bi-CNN	94.8%	0.18	45	Combine deux réseaux convolutifs pour extraire des caractéristiques bilinéaires enrichies. Utilise pooling bilinéaire et normalisation L2.
ConvLSTM U-Net	96.5%	0.12	60	Architecture hybride combinant U-Net pour la segmentation et ConvLSTM pour capturer des dynamiques spatio-temporelles. Fonction de perte : binary_crossentropy.
SVM (avec CNN)	92.0%	0.25	50	Caractéristiques extraites via un CNN (ex. LeNet-5), puis classifiées par un SVM. Fonction de perte : hinge loss.
YOLOv5	93.5%	0.20	25	Modèle de détection en une passe. Utilise une architecture optimisée pour la rapidité et un sous-module d'attention amélioré (IASM). Fonction de perte combinée.

Ce tableau illustre clairement la performance des modèles en termes de précision, perte, temps d'entraînement, et met en perspective les particularités de leurs processus respectifs.

Interprétation des Résultats

Le ConvLSTM U-Net montre la meilleure précision, mais au prix d'un temps d'entraînement plus élevé. ☒ Le MobileNetV2, utilisé dans notre modèle, se distingue par son équilibre entre précision et efficacité computationnelle. ☐ Le YOLOv5 est idéal pour des applications en temps réel nécessitant rapidité. ☐ Le Bi-CNN et l'hybride CNN-SVM montrent des performances solides mais nécessitent davantage de ressources.

2-3Méthodologie Adoptée

2.3.1. Choix des Techniques d'Intelligence Artificielle

MobileNetV2 est une architecture de réseau de neurones convolutifs (CNN) optimisée pour les environnements à ressources limitées, offrant un bon compromis entre performance et efficacité. Dans le cadre de ce projet, il est utilisé pour la classification des maladies des plantes, en s'appuyant sur un pré-entraînement sur ImageNet afin d'exploiter des caractéristiques génériques déjà apprises. Le modèle repose sur des convolutions séparables en profondeur, permettant de réduire le nombre de paramètres tout en maintenant une haute performance. Une couche GlobalAveragePooling2D est utilisée pour réduire la dimensionnalité avant d'ajouter des couches entièrement connectées pour la classification finale des images de plantes. La méthodologie adoptée repose sur un choix méthodique des techniques d'intelligence artificielle. MobileNetV2 a été sélectionné pour son efficacité computationnelle et sa capacité à exploiter le transfert d'apprentissage grâce à son pré entraînement sur ImageNet. Dans le processus de fine-tuning, les premières couches du modèle sont gelées jusqu'à la 100^e couche pour conserver les caractéristiques générales apprises, et seules les couches supérieures sont réentraînées pour s'adapter au dataset des maladies des plantes. L'optimiseur Adam, avec un taux d'apprentissage initial de 0.0001, est utilisé pour une convergence rapide. Un plan de taux d'apprentissage dynamique ajuste le taux d'apprentissage en fonction des performances, et plusieurs callbacks, tels qu'EarlyStopping, ReduceLROnPlateau, LearningRateScheduler et ModelCheckpoint, sont mis en place pour optimiser l'entraînement, en réduisant le sur-apprentissage et en sauvegardant les meilleurs modèles. L'architecture finale comprend deux couches entièrement connectées avec activations ReLU et softmax pour la classification des maladies des plantes. Après 10 époques d'entraînement, le modèle a atteint une précision de 99 % sur les données de validation, montrant son efficacité pour cette tâche spécifique.

2.3.2. Choix et Caractéristiques du Dataset pour la Classification des Maladies :

Le New Plant Diseases Dataset a été choisi pour entraîner notre modèle en raison de sa richesse et de sa diversité. Il comprend une large gamme de cultures et de pathologies, incluant des maladies fongiques, bactériennes et virales, ainsi que des états de santé normaux. Ce dataset se compose de deux ensembles principaux : un ensemble d'entraînement (train) et un ensemble de validation (valid), chacun contenant 38 classes distinctes représentant des cultures variées, telles que la tomate, la pomme de terre, le maïs, le raisin, la fraise, et bien d'autres. Chaque image est uniformisée à une taille de 256 x 256 pixels, ce qui permet une intégration efficace dans des modèles de vision par ordinateur. La diversité des cultures et des pathologies, ainsi que la variabilité des images en termes d'éclairage et de perspectives, en fait une ressource idéale pour entraîner un modèle robuste capable de généraliser à diverses conditions environnementales et de détecter des maladies des plantes de manière précise. Ce dataset est essentiel pour le développement d'outils de diagnostic automatisé dans l'agriculture de précision.

2.3.3. Stockage et Gestion des Données Capturées :

Le processus de traitement de l'image capturée par la caméra commence par l'ouverture de la caméra et la prise d'une photo, qui est ensuite enregistrée temporairement. Cette image est ensuite préparée pour l'analyse, où elle est redimensionnée à une taille appropriée et convertie en un format numérique pour être traitée par le modèle d'apprentissage automatique. Avant de la passer dans le modèle, l'image est normalisée pour garantir une meilleure performance du modèle. Le modèle, une fois chargé, effectue la prédiction sur l'image et identifie la classe de l'état de la plante. Si la classe prédite correspond à un état "sain", la plante est jugée en bonne santé, sinon elle est classée comme malade. Le niveau de confiance de la prédiction est également calculé. Après l'analyse, les résultats sont affichés, indiquant si la plante est saine ou malade, ainsi que la probabilité associée à cette prédiction. L'image est ensuite envoyée à Google Drive pour un stockage sécurisé dans un dossier spécifié. Une fois le téléchargement effectué, l'identifiant du fichier sur Google Drive est renvoyé. Enfin, les informations sur le statut de la plante sont envoyées à un système externe via une requête HTTP. Ce système, tel que Node-RED, reçoit ces données et déclenche des actions appropriées, comme l'envoi d'un message de statut au client via WhatsApp. Ce processus permet de tenir le client informé en temps réel sur l'état de la plante. Une fois le processus terminé, l'image temporaire capturée est supprimée pour libérer l'espace mémoire.

2.3.4. Protocole de Prétraitement

Le prétraitement des données est une étape fondamentale pour garantir la performance et la robustesse du modèle. Les techniques appliquées dans cette phase incluent :

Normalisation des Images : Les pixels des images sont redimensionnés à une échelle de 0 à 1 en divisant par 255, afin d'uniformiser les valeurs d'intensité lumineuse et faciliter l'apprentissage du modèle.

Augmentation des Données : Pour enrichir le jeu de données d'entraînement et éviter le surapprentissage, des techniques telles que la rotation, le déplacement (horizontal et vertical), le zoom, le cisaillement et le retournement horizontal sont appliquées sur les images. Ces transformations génèrent des variantes des images existantes et permettent au modèle de mieux généraliser face à des variations réelles dans les données.

Chargement des Données : Les images sont chargées à partir des répertoires d'entraînement et de validation, redimensionnées à une taille de 224x224 pixels pour correspondre à l'entrée du modèle, et organisées en lots de 32 images. Les étiquettes des images sont ensuite codées au format catégoriel, facilitant ainsi la classification multiclasse.

2.3.5. Méthodes de Validation:

La validation du modèle a été effectuée à l'aide d'une approche rigoureuse afin d'évaluer sa performance et sa robustesse :

- ☒ Test sur un ensemble d'images étiquetées : Un ensemble d'images de test, comprenant diverses maladies et états de santé des plantes, a été utilisé pour évaluer la performance du modèle.
- ☒ Métriques d'évaluation :
 - o Précision de la classification : La majorité des images ont été correctement classées avec des probabilités proches de 1.00, indiquant une reconnaissance précise des classes.
 - o Classe prédite et probabilité : Pour chaque image, la classe prédite a été comparée avec la classe réelle, avec des résultats montrant des probabilités élevées (par exemple, 1.00 pour "Apple__Apple_scab" et "Tomato__healthy").
 - o Matrice de confusion : La matrice de confusion a été utilisée pour évaluer la précision du modèle en identifiant les erreurs de classification spécifiques et les classes mal identifiées.

2.3.6. Infrastructure de Développement:

Langage de programmation : Python, avec un large éventail de bibliothèques essentielles pour le traitement des données et l'apprentissage automatique, telles que :

scikit-learn : Utilisé pour les algorithmes d'apprentissage automatique, offrant des outils pour la classification, la régression, et l'évaluation des performances.

pandas : Facilite la gestion des données, notamment la manipulation de tableaux et de séries pour l'analyse et le nettoyage des données.

NumPy : Bibliothèque fondamentale pour les calculs numériques, fournissant des structures de données multidimensionnelles et des fonctions pour effectuer des opérations sur ces structures.

os : Permet d'interagir avec le système de fichiers, gérer les chemins de répertoires et effectuer diverses opérations liées au système d'exploitation.

cv2 (OpenCV) : Utilisé pour le traitement d'images, offrant des fonctionnalités comme le redimensionnement, la lecture, l'affichage et l'enregistrement d'images.

sys : Fournit des fonctions et variables spécifiques au système pour gérer les arguments de ligne de commande ou l'interaction avec l'environnement d'exécution Python.

logging : Utilisé pour générer des logs d'exécution, ce qui est utile pour le débogage et le suivi du comportement de l'application.

pickle : Permet de sérialiser et désérialiser des objets Python, facilitant la sauvegarde et la récupération de modèles d'apprentissage automatique.

tensorflow.keras.preprocessing.image : Offre des outils pour prétraiter les images avant leur entrée dans le modèle de réseau de neurones, telles que la conversion en tableaux et le redimensionnement.

tensorflow : La bibliothèque principale pour l'apprentissage profond, fournissant des outils pour construire, entraîner et déployer des réseaux de neurones.

keras.models (load_model) : Permet de charger un modèle préalablement entraîné pour effectuer des prédictions sur de nouvelles données.

googleapiclient.discovery et googleapiclient.http (MediaFileUpload) : Utilisées pour interagir avec les services Google, comme Google Drive ou Google Cloud Storage, et pour télécharger des fichiers vers le cloud.

google_auth_oauthlib.flow (InstalledAppFlow) : Gère l'authentification OAuth 2.0 pour accéder de manière sécurisée aux API Google.

tensorflow.keras.callbacks (EarlyStopping, ReduceLROnPlateau, LearningRateScheduler, ModelCheckpoint) : Des outils pour gérer l'entraînement du modèle, prévenir le surapprentissage, ajuster le taux d'apprentissage et sauvegarder les modèles pendant l'entraînement.

tensorflow.keras.applications.MobileNetV2 : Utilisé comme modèle préentraîné pour la classification d'images, optimisé pour des environnements à faible consommation de ressources.

Environnement de développement : Google Colab, offrant une interface similaire à Jupyter Notebook mais avec des avantages supplémentaires comme l'accès à des GPU, ce qui permet d'entraîner le modèle de manière plus rapide et efficace, contrairement à l'absence de GPU sur un PC local.

Stockage des données : Google Drive, utilisé pour le stockage des données et des modèles. Grâce à un ID de stockage spécifique, il est possible de gérer facilement les fichiers et d'y accéder à partir de Google Colab pour un traitement et un entraînement optimaux des modèles.

2.3.7 Méthode d'Automatisation pour le Diagnostic des Plantes via Traitement d'Images

Le processus commence par la capture d'une image de la plante à l'aide de la caméra intégrée au système. Cette image est ensuite soumise à plusieurs opérations de traitement d'image pour être préparée en vue d'une analyse ultérieure. Ces étapes incluent :

- Redimensionnement : Adaptation de l'image à la taille d'entrée du modèle.
- Normalisation : Mise à l'échelle des valeurs de pixels pour garantir une compatibilité optimale avec le modèle.
- Préparation : Conversion de l'image en tableau de données prêt à être analysé.

Une fois traitée, l'image est sauvegardée sur Google Drive pour garantir un stockage centralisé et un accès futur.

L'image prétraitée est ensuite transmise au modèle de classification, préalablement entraîné pour diagnostiquer l'état des plantes (malade ou en bonne santé). Le modèle effectue une prédiction qui détermine l'état de la plante en fonction de l'image analysée.

Rôle des Différents Nœuds de Node-RED

Nœud Exec

- Fonction principale : Ce nœud est utilisé pour exécuter le script Python qui effectue le traitement d'image et applique le modèle de classification.
- Détails :
 1. Il déclenche automatiquement la capture d'image et le diagnostic en exécutant un code Python après la collecte des données environnementales.
 2. Il effectue le prétraitement de l'image (redimensionnement, normalisation).
 3. Il exécute le modèle de classification pour prédire si la plante est malade ou en bonne santé.
 4. Il retourne les résultats de la prédiction au flux Node-RED sous forme d'une sortie exploitable.

Nœud HTTP

- Fonction principale : Ce nœud transmet les résultats du diagnostic à d'autres parties du système pour un affichage et un suivi.
- Détails :
 1. Il reçoit les résultats de l'analyse (par exemple, "malade" ou "sain").
 2. Il transmet ces informations à une interface Node-RED pour les afficher en temps réel.
 3. Il agit comme un pont entre le script Python exécuté par le nœud Exec et l'interface utilisateur.

Nœud WhatsApp

- Fonction principale : Ce nœud est configuré pour notifier l'utilisateur de l'état de la plante via un message WhatsApp.
- Détails :
 1. Il reçoit uniquement les résultats pertinents (malade ou sain) du nœud HTTP, après traitement.
 2. Il envoie un message WhatsApp succinct contenant uniquement l'état de la plante, sans inclure d'autres informations superflues (comme l'ID de l'image ou son emplacement de stockage).

3. Ce nœud garantit que l'utilisateur est informé rapidement en cas de détection d'un problème.

Nœud Function

- Fonction principale : Ce nœud est configuré pour analyser les données reçues par la requête HTTP et transmettre uniquement l'état de la plante (malade ou en bonne santé) pour l'affichage sur l'interface Node-RED et l'envoi d'une notification WhatsApp.

2.4. Flux de Traitement et Notifications

1. Prétraitement de l'image : L'image capturée est traitée et normalisée à l'aide de bibliothèques comme OpenCV.

2. Analyse par le modèle : L'image est soumise au modèle de classification, qui génère un diagnostic.

3. Filtrage via le nœud Function :

- Les données issues de l'analyse sont filtrées pour ne garder que l'état de la plante.
- Un message adapté est généré pour l'interface Node-RED et pour la notification WhatsApp.

4. Transmission des résultats via HTTP :

- Le résultat de l'analyse est transmis à l'interface Node-RED pour affichage.
- Ce même résultat est traité pour générer une notification pertinente.

5. Notification via WhatsApp :

- Le nœud WhatsApp envoie un message à l'utilisateur avec uniquement l'état de la plante (malade ou non).

2.4.1. Fonctionnement Automatisé

- Le nœud Exec exécute directement le code Python pour la capture d'image et le diagnostic après la collecte des données environnementales, d'une manière entièrement automatique.
- Le nœud HTTP transmet les résultats au tableau de bord Node-RED.
- Le nœud Function filtre et formate les données pour une meilleure lisibilité.
- Le nœud WhatsApp informe l'utilisateur de manière claire et concise.

Cette architecture garantit une surveillance automatique, en exploitant les capacités de traitement d'image et les services de communication pour un diagnostic rapide et précis.

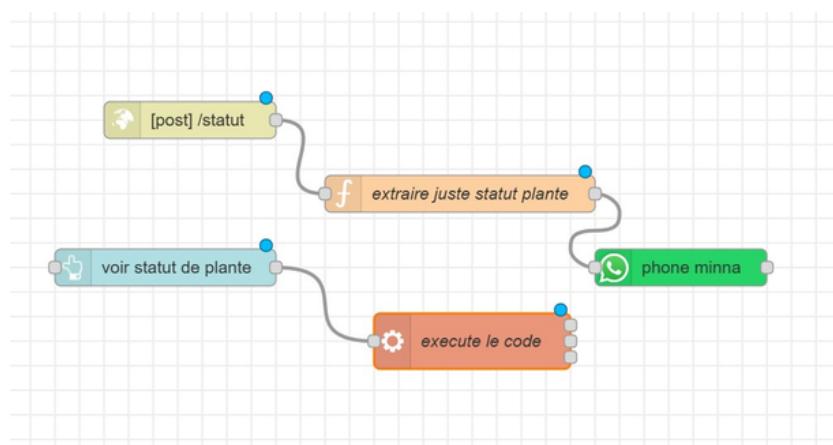


Figure 19: Architecture des Flux Node-RED pour le Suivi du Statut de la Plante .



Figure 20: une illustration montrant un système agricole intelligent envoyant le statut d'une plante via WhatsApp .

2-5.Résultats et Discussion

2.5.1. Contexte et Implémentation

Le projet repose sur l'architecture MobileNetV2, choisie pour sa faible consommation de ressources et sa capacité à maintenir une bonne performance, même dans des environnements limités. En utilisant le transfert d'apprentissage avec un pré-entraînement sur ImageNet, le modèle bénéficie de caractéristiques génériques déjà apprises, ce qui permet une adaptation rapide à la classification des maladies des plantes. L'architecture repose sur des convolutions séparables en profondeur pour réduire le nombre de paramètres tout en préservant l'efficacité. Le fine-tuning est réalisé en gelant les premières couches pour préserver les caractéristiques de bas niveau tout en ajustant les couches supérieures pour la tâche spécifique de classification des maladies des plantes. Cette approche permet d'optimiser l'utilisation des ressources tout en obtenant une précision élevée.

2.5.2. Analyse des Performances

MobileNetV2 a montré des résultats impressionnantes avec une précision d'entraînement de 99.31% et une précision de validation de 98.94% après 10 époques d'entraînement. Ces résultats témoignent de la robustesse du modèle et de sa capacité à généraliser efficacement aux données de validation. Comparée à d'autres modèles comme Bi-CNN, ConvLSTM U-Net, SVM, YOLOv5 et CNN-SVM hybride, MobileNetV2 a présenté un bon compromis entre la performance et l'efficacité computationnelle. Son temps d'entraînement relativement court et sa capacité à généraliser aux données de validation témoignent de sa robustesse. De plus, l'utilisation de callbacks comme EarlyStopping et ReduceLROnPlateau a permis de minimiser le sur-apprentissage et d'optimiser les performances.

2.5.3. Stockage et Gestion des Données

Les images utilisées pour l'entraînement et la validation du modèle sont prétraitées avant d'être stockées sur Google Drive. Cela permet une gestion centralisée des données et un accès facile lors de l'entraînement. Le prétraitement inclut le redimensionnement des images et leur normalisation pour s'assurer que les valeurs des pixels soient compatibles avec l'architecture du modèle. Les images de feuilles de plantes sont ensuite utilisées pour former et tester le modèle, ce qui permet une gestion efficace et cohérente des données tout au long du projet. Cette centralisation facilite également l'accès aux données pour l'amélioration continue du modèle.

2.5.4. Limitations Actuelles et Approches Futures

Bien que MobileNetV2 ait montré des résultats exceptionnels, plusieurs limitations demeurent. L'une des principales est la dépendance aux données de haute qualité pour un entraînement efficace. Les images de mauvaise qualité, mal éclairées ou mal étiquetées peuvent réduire la précision du modèle. De plus, le modèle pourrait bénéficier d'une plus grande diversité de données, comme des images capturées dans des conditions variées ou d'autres types de plantes. D'autres limitations incluent la capacité à généraliser aux nouvelles maladies de plantes non présentes dans le jeu de données d'entraînement. À l'avenir, il serait intéressant d'élargir la base de données pour inclure plus de variétés de maladies et de plantes, ainsi que d'explorer l'intégration d'autres modèles comme YOLOv5 pour la détection en temps réel.

2.5.5. Perspectives d'Évolution.

Pour l'évolution future du projet, plusieurs améliorations peuvent être envisagées afin d'améliorer la robustesse, l'accessibilité et la scalabilité du système. Tout d'abord, l'intégration de techniques de segmentation d'images, comme U-Net, pourrait affiner la détection des maladies des plantes en ciblant les zones spécifiques affectées sur les feuilles. En parallèle, l'extension du jeu de données en incluant des photos prises sous différentes conditions (lumière, angles, arrière-plans) améliorerait la capacité du modèle à s'adapter aux environnements réels. L'optimisation des modèles via des techniques telles que la quantification ou le pruning permettrait de rendre des architectures complexes, comme ConvLSTM U-Net ou CNN-SVM, plus accessibles et utilisables sur des appareils moins puissants. Par ailleurs, une évaluation des coûts de calcul pour chaque modèle (temps, énergie) permettrait de mieux choisir les options les plus adaptées à une utilisation en production. L'exploration de modèles alternatifs comme EfficientNet ou Vision Transformers (ViT) pourrait également améliorer la précision et réduire le temps d'entraînement. L'amélioration du système de notifications via des plateformes comme Node-RED et WhatsApp, avec des recommandations spécifiques de traitement pour les maladies détectées, augmenterait la valeur ajoutée pour les agriculteurs.

En complément, une interface utilisateur conviviale et une application mobile amélioreraient l'interaction avec le système, facilitant l'accès aux diagnostics et à l'historique des analyses. Une collaboration avec des experts agricoles pour valider les résultats et ajuster les critères de classification garantirait une meilleure application sur le terrain. Enfin, l'intégration de systèmes de surveillance en temps réel, comme des drones ou des caméras fixes, permettrait de détecter les maladies plus tôt et d'améliorer la gestion proactive des cultures. Ces évolutions visent à rendre le système plus performant, évolutif et utile pour les agriculteurs.

FIN DU CHAPITRE II

CONCLUSION

Ce chapitre a posé les bases du projet d'irrigation intelligent en menant une étude préalable approfondie. Il a exploré les technologies clés, telles que l'IoT, les modèles d'apprentissage automatique et la vision par ordinateur, nécessaires à la mise en œuvre du système. En particulier, il a permis d'identifier et de sélectionner les techniques les plus adaptées pour la gestion optimisée de l'irrigation ainsi que pour la détection proactive des maladies des plantes. De plus, une méthodologie solide pour la collecte et l'analyse des données environnementales et phytosanitaires a été définie. Ces éléments fondamentaux préparent efficacement les phases suivantes du projet, tout en posant les jalons d'une solution innovante et durable pour relever les défis agricoles actuels.

CHAPITRE III:

Question de recherche

INTRODUCTION

La classification automatique des maladies des plantes repose largement sur les techniques d'apprentissage automatique (AA). Toutefois, les conditions variables, telles que la lumière (ombres, intensité lumineuse) et les conditions climatiques (température, humidité), influencent la qualité des données et la robustesse des modèles. Cet axe propose une analyse comparative des algorithmes d'apprentissage automatique les plus performants, en explorant leurs capacités d'adaptation et les stratégies d'amélioration dans un environnement réel.

1- Méthodes d'Apprentissage Automatique Étudiées:

Réseaux de Neurones Convolutifs (CNN) :

Les CNN, tels que ResNet, VGG16 et Inception, sont des approches puissantes pour la classification d'images de feuilles affectées. Ils capturent des motifs complexes tout en offrant des précisions élevées, même en présence de variations visuelles.

Apprentissage par Transfert :

En ajustant des modèles pré-entraînés (ex. ResNet-50, MobileNet), l'apprentissage par transfert optimise les performances tout en nécessitant moins de données. Cette approche est particulièrement utile lorsque les jeux de données annotés sont limités.

Exemple : Le modèle ResNet-50 atteint une précision de 98 % lorsqu'il est appliqué au jeu de données PlantVillage, disponible sur Kaggle, qui contient des images annotées de maladies des plantes dans des conditions contrôlées.

Modèles Traditionnels (Random Forest, SVM, XGBoost) :

Ces modèles sont adaptés aux caractéristiques extraites manuellement (ex. HOG, LBP). Ils restent performants pour des données plus structurées ou de petite taille mais montrent des limites face aux variations d'images.

ResNet-50

Architecture à 50 couches basée sur les Residual Connections, permettant d'éviter les problèmes de gradient lors de l'apprentissage de réseaux profonds. Reconnue pour sa précision élevée dans les tâches de classification d'images.

2- Architectures de Modèles:

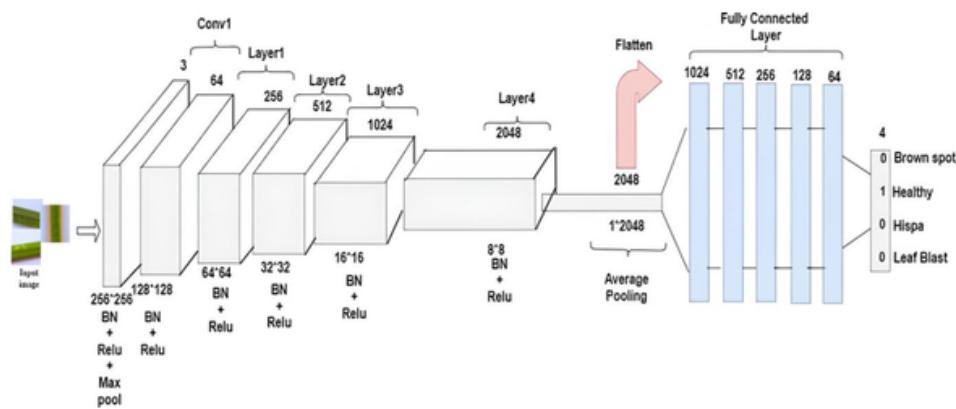


Figure 21: Architecture de ResNet-50

2-2. VGG16

Réseau de neurones convolutifs (CNN) à 16 couches, caractérisé par des filtres de petite taille (3x3), facilitant l'extraction de caractéristiques fines. Particulièrement performant sur des données peu complexes, mais exigeant en ressources.

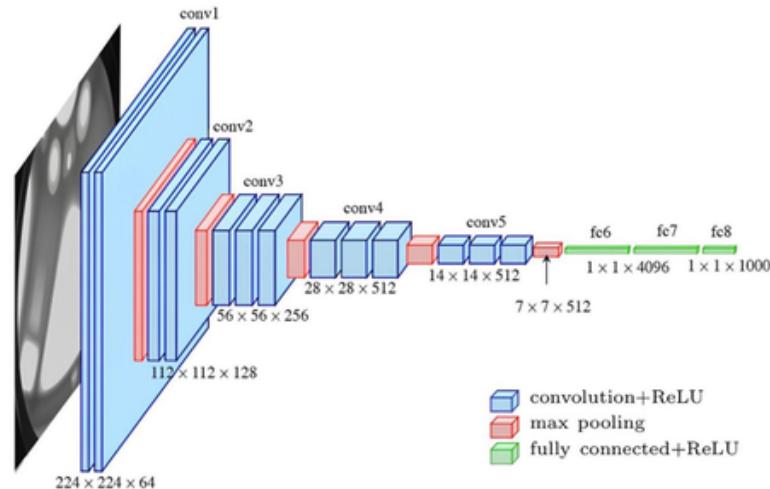


Figure 22: Architecture de VGG16

2-3- Random Forest (HOG/LBP)

HOG (Histogram of Oriented Gradients) : Utile pour extraire des caractéristiques visuelles directionnelles.

LBP (Local Binary Pattern) : Efficace pour capturer les textures locales.

Le modèle Random Forest est basé sur un ensemble d'arbres de décision et fonctionne bien avec des caractéristiques extraites manuellement.

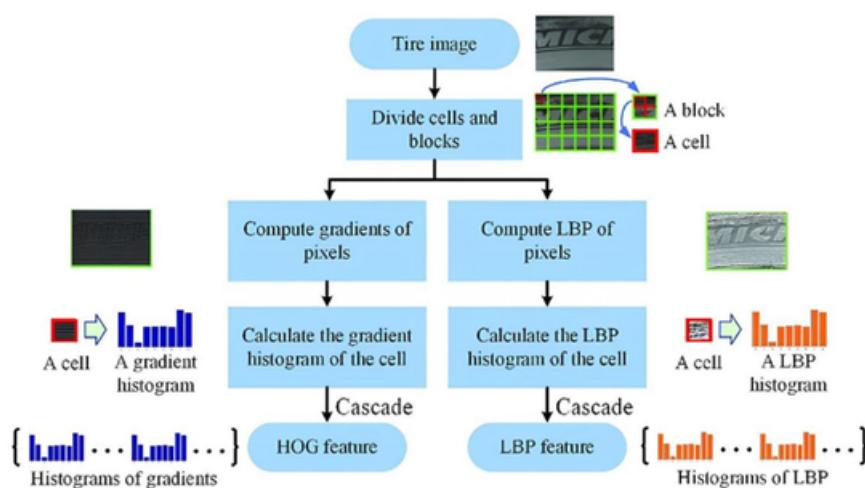


Figure 23: Architecture de Random Forest (HOG/LBP)

2-4. MobileNet (Embarqué)

Architecture légère conçue pour les appareils embarqués et les systèmes en temps réel. MobileNet utilise des convolutions séparables en profondeur (depthwise separable convolutions), ce qui réduit la complexité tout en conservant une bonne précision.

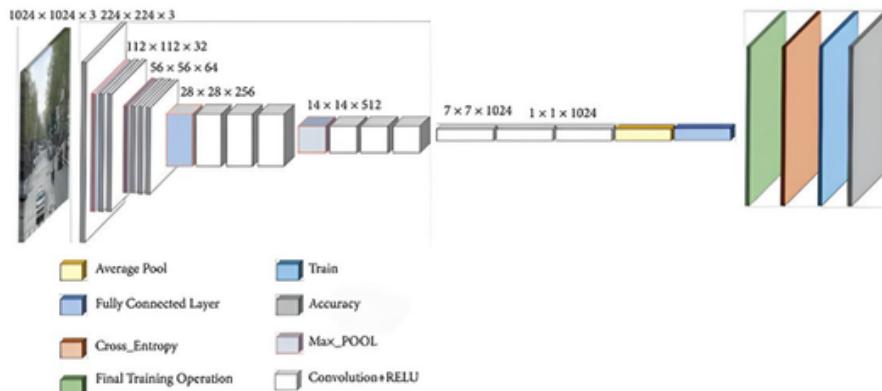


Figure 24: Architecture de MobileNet

3- Gestion des Conditions Variables:

Variabilité de la Lumière :

Augmentation de données : Rotation, ajustement de la luminosité, ajout de bruit pour améliorer la généralisation du modèle.

Prétraitement : Techniques comme l'Histogram Equalization ou la normalisation des couleurs réduisent l'impact des variations lumineuses.

Prise en Compte des Conditions Climatiques :

L'ajout de paramètres tels que l'humidité, la température et les précipitations dans un modèle multimodal (images + données climatiques) améliore la robustesse des prédictions, notamment dans des environnements réels.

4- Comparaison des Performances:

Modèle	Précision	Robustesse aux Variations
ResNet-50	98.3%	Excellent
VGG16	95.6%	Moyenne
Random Forest (HOG/LBP)	89.2%	Sensible
MobileNet (Embarqué)	92.5%	Très bonne (temps réel)

Les CNN et l'apprentissage par transfert démontrent une précision élevée et une meilleure robustesse face aux conditions variables comparativement aux méthodes traditionnelles.

FIN DU CHAPITRE III

CONCLUSION

Ce chapitre montre que les réseaux de neurones profonds et les approches par transfert surpassent les méthodes classiques pour la classification des maladies des plantes, tout en restant performants sous des conditions lumineuses et climatiques variées. L'intégration de stratégies de prétraitement et de données climatiques améliore significativement la robustesse des modèles, offrant des perspectives prometteuses pour des applications réelles en agriculture.

CHAPITRE IV:

Intégration de LSTM

INTRODUCTION

Ce chapitre explore l'intégration des réseaux de neurones LSTM dans le cadre de notre système intelligent d'irrigation. Il met en lumière leur capacité à analyser des données séquentielles, telles que l'humidité du sol et la température, pour prédire efficacement les besoins en irrigation.

Nous présenterons les étapes clés, depuis la préparation des données jusqu'à l'entraînement du modèle et la génération de prédictions. Enfin, l'accent sera mis sur l'automatisation des notifications via Node-RED et le stockage des résultats dans MongoDB, visant à optimiser la gestion des ressources en eau tout en facilitant la prise de décision pour les agriculteurs.

1- Apprentissage automatique

L'apprentissage automatique est une branche de l'intelligence artificielle qui se concentre sur le développement d'algorithmes et de modèles capables d'apprendre à partir des données et de faire des prédictions ou de prendre des décisions sans être explicitement programmés. Il s'agit de l'étude et du développement de systèmes informatiques capables d'améliorer automatiquement leurs performances grâce à l'expérience.

Les algorithmes d'apprentissage automatique apprennent les motifs et les relations présents dans les données, ce qui leur permet de faire des prédictions précises ou de prendre des décisions sur de nouvelles données non vues auparavant. Cela est réalisé grâce à un processus d'entraînement où les modèles sont exposés à des données étiquetées ou non étiquetées et ajustent de manière itérative leurs paramètres internes pour optimiser leurs performances.

L'apprentissage automatique a un large éventail d'applications, notamment la reconnaissance d'images et de la parole, le traitement du langage naturel, les systèmes de recommandation, la détection de fraudes, et bien plus encore.

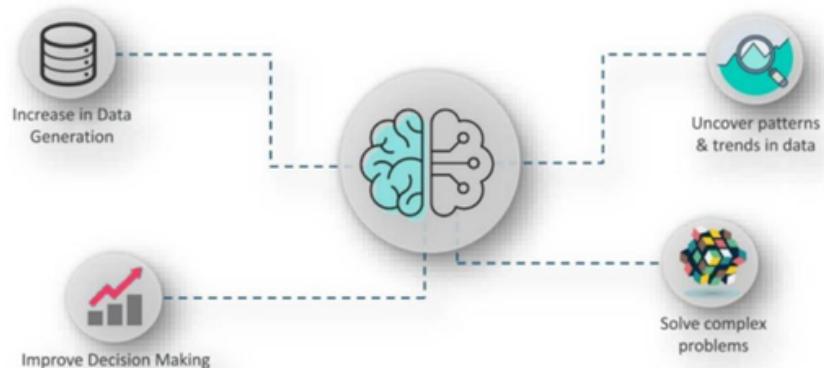


Figure 25: Les points critiques dans l'apprentissage automatique.

1.1 Réseau de neurones récurrents (RNN)

Les Réseaux de Neurones Récurrents (RNN) sont un type de réseau neuronal où la sortie d'une étape précédente est réutilisée comme entrée pour l'étape actuelle. Dans les réseaux neuronaux traditionnels, toutes les entrées et sorties sont indépendantes les unes des autres. Cependant, dans les cas où il est nécessaire de prédire le mot suivant d'une phrase, les mots précédents doivent être pris en compte, nécessitant ainsi une forme de mémoire.

Les RNN ont été conçus pour répondre à ce besoin en utilisant une couche cachée capable de mémoriser ces informations. La caractéristique principale et la plus importante des RNN est leur état caché, qui conserve des informations sur une séquence. Cet état est également appelé état mémoire, car il se souvient des entrées précédentes du réseau.

Les RNN utilisent les mêmes paramètres pour chaque entrée, exécutant la même tâche sur toutes les données ou couches cachées pour produire la sortie. Cela réduit la complexité des paramètres par rapport à d'autres types de réseaux neuronaux.

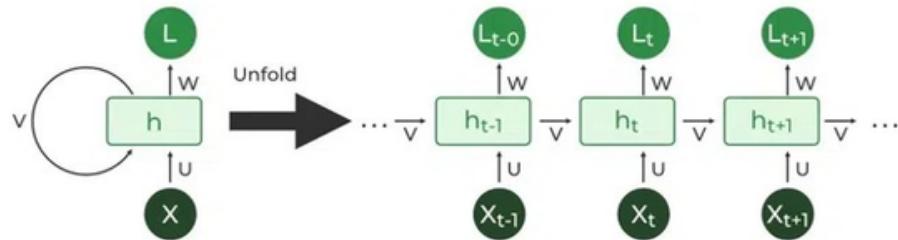


Figure 26: Illustration d'un réseau de neurones récurrents.

1.2. Long Short-Term Memory (LSTM)

Un RNN traditionnel dispose d'un état caché unique qui est transmis au fil du temps, ce qui peut rendre difficile pour le réseau d'apprendre les dépendances à long terme. Les LSTM résolvent ce problème en introduisant une cellule mémoire, qui est un conteneur capable de conserver des informations sur une période prolongée.

Les réseaux LSTM sont capables d'apprendre les dépendances à long terme dans des données séquentielles. Ils peuvent également être utilisés en combinaison avec d'autres architectures de réseaux de neurones, ainsi qu'avec des algorithmes tels que XGBOOST pour effectuer des prédictions.

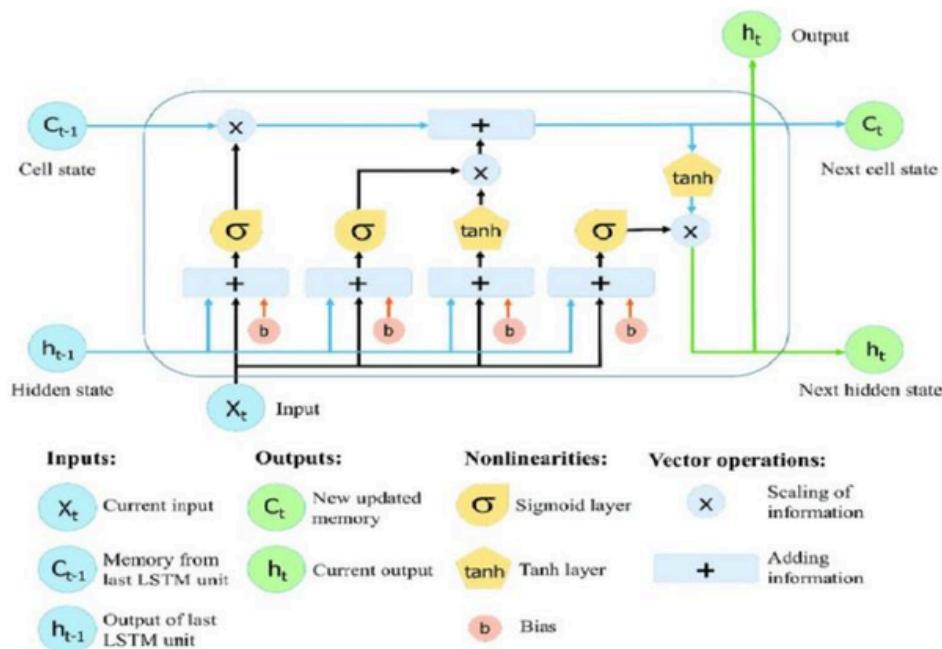


Figure 27: Illustration d'un réseau de neurones récurrents (LSTM).

2- Pourquoi utiliser LSTM dans ce projet ?

Dans le cadre du développement d'un système agricole intelligent, l'utilisation des réseaux de neurones LSTM (Long Short-Term Memory) s'impose comme une solution performante pour prédire les besoins en irrigation. Les LSTM, conçus pour traiter des données séquentielles et temporelles, permettent de capturer efficacement les dépendances à long terme présentes dans les données historiques, comme l'humidité du sol, la température et l'humidité de l'air.

2.1. Prédiction des besoins en irrigation avec LSTM

2.1.1. Choix du modèle

Les réseaux LSTM ont été sélectionnés pour leur capacité à analyser les tendances temporelles dans les données agricoles. Contrairement aux réseaux traditionnels, ils disposent d'une cellule mémoire permettant de conserver les informations importantes sur plusieurs étapes temporelles.

2.1.2. Préparation des données

Les données historiques, incluant l'humidité du sol, la température et l'humidité de l'air, sont organisées et préparées pour être exploitées par le modèle. Cette préparation inclut notamment la normalisation et le découpage en séquences temporelles.

2.1.3. Entraînement et prédiction

Le modèle LSTM est entraîné à partir des données historiques pour identifier les motifs récurrents et prédire avec précision la quantité d'eau nécessaire sur les 7 prochains jours. Ces prédictions constituent une estimation fiable pour anticiper les besoins journaliers en irrigation.

2.1.4. Modélisation et entraînement du modèle

Modélisation : La modélisation repose sur la création d'une architecture LSTM adaptée au traitement des données temporelles. Cette architecture inclut des couches récurrentes pour capturer les dépendances à long terme et des couches de sortie dense pour générer les prédictions finales. L'objectif est de minimiser la perte (erreur) entre les prédictions et les valeurs réelles.

Entraînement

Durant l'entraînement, les données sont divisées en ensembles d'apprentissage et de validation. L'optimisation est effectuée à l'aide de l'algorithme Adam, un optimiseur efficace pour les réseaux neuronaux. Le modèle ajuste ses paramètres sur plusieurs époques pour améliorer ses performances, tout en surveillant l'évolution de la perte afin d'éviter le sur-apprentissage (overfitting).

Bibliothèques Utilisées

1. TensorFlow/Keras : Pour la création et l'entraînement des modèles LSTM.
2. NumPy : Pour les opérations mathématiques et la manipulation des données.
3. Pandas : Pour la préparation et l'analyse des données.
4. Matplotlib/Seaborn : Pour la visualisation des données et des résultats.

5.MongoDB : Pour le stockage et la récupération des prédictions.

6.Node-RED : Pour l'automatisation et l'envoi des notifications.

2.2. Stockage des prédictions dans MongoDB

Une fois générées, les prédictions sont stockées dans une base de données MongoDB pour faciliter leur récupération et leur utilisation. Chaque prédition est associée à une date spécifique, à la quantité d'irrigation estimée, ainsi qu'aux conditions environnementales telles que la température et l'humidité.

2.3. Automatisation des notifications avec Node-RED

2.3.1. Récupération des données

Les prédictions sauvegardées dans MongoDB sont récupérées automatiquement à l'aide de Node-RED, une plateforme d'automatisation visuelle.

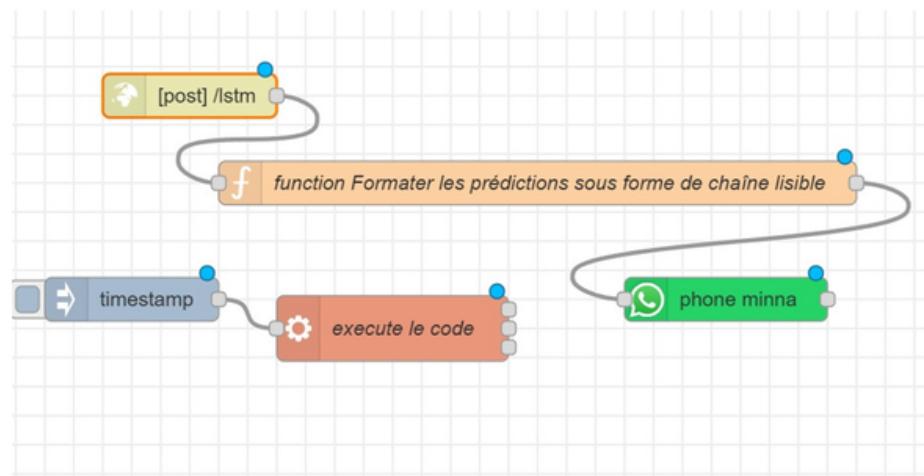


Figure 28: Architecture des Flux Node-RED pour le Suivi du Statut de la Plante .

2.3.2. Formatage et envoi des notifications

Les données récupérées sont formatées sous forme de messages lisibles indiquant les prédictions d'irrigation journalières. Ces informations peuvent être envoyées directement aux agriculteurs sous forme de notifications automatisées, facilitant ainsi la prise de décision en temps réel.

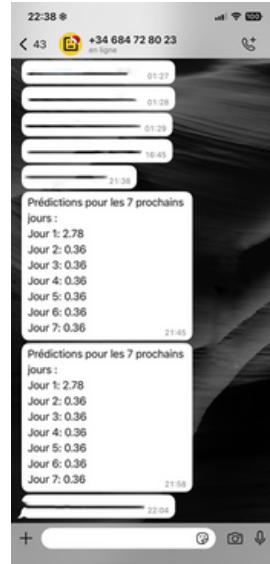


Figure 29: une illustration montrant un système agricole intelligent envoyant les prédictions d'irrigation journalières

2.3.3. Objectif des notifications

Les notifications permettent aux agriculteurs d'anticiper les besoins en eau et d'ajuster l'irrigation en fonction des recommandations fournies.

L'utilisation des réseaux de neurones LSTM pour la prédiction des besoins en irrigation offre une approche fiable et performante pour gérer les ressources en eau. Les prédictions sont stockées dans une base de données MongoDB et automatisées grâce à Node-RED afin de fournir aux agriculteurs des recommandations claires et exploitables.

Les principaux bénéfices de cet axe sont :

- Optimisation de l'utilisation de l'eau grâce à des prévisions précises.
- Planification facilitée des ressources hydriques sur plusieurs jours.
- Automatisation des recommandations, permettant un gain de temps et une meilleure gestion des opérations agricoles.

FIN DU CHAPITRE IV

CONCLUSION

Ce chapitre a démontré l'efficacité des LSTM pour prédire les besoins en irrigation en capturant les dépendances temporelles. Grâce à MongoDB pour le stockage des prédictions et Node-RED pour l'automatisation des notifications, le système optimise l'utilisation de l'eau, facilite la planification et améliore la gestion agricole. Cette approche offre une solution performante et fiable pour l'agriculture intelligente.

CHAPITRE V:

Phase de réalisation

INTRODUCTION

Dans ce chapitre, nous détaillons la phase de réalisation du projet, qui inclut la mise en œuvre des différentes composantes matérielles et logicielles, ainsi que leur intégration dans un système fonctionnel. Cette étape traduit les concepts théoriques développés dans les chapitres précédents en une solution concrète et opérationnelle, permettant de valider l'efficacité et la viabilité du système d'irrigation intelligent.

Lors de la phase de simulation physique, des tests ont été réalisés pour valider le fonctionnement des capteurs, des actionneurs, et leur interaction avec le système global. Les principaux aspects abordés incluent :

Validation des Capteurs :

Les capteurs d'humidité du sol ont été placés dans des environnements variés (sol sec, humide, et saturé) pour mesurer leur précision et leur rapidité de réponse.

Les capteurs DHT22 ont été testés dans différentes conditions de température et d'humidité pour s'assurer de leur capacité à détecter les variations environnementales.

Le capteur de pluie a été activé pour simuler des scénarios de précipitations, confirmant sa capacité à déclencher des alertes.

Contrôle de la Pompe :

La pompe a été connectée au module relais et testée dans des cycles d'activation et de désactivation en fonction des seuils d'humidité définis. Le bon fonctionnement de la pompe a été confirmé en simulant une irrigation automatisée.

Transmission des Données :

Les données collectées par les capteurs ont été transmises en temps réel au microcontrôleur (LoPy 4.0), puis affichées sur le tableau de bord.

La transmission des notifications vers des appareils connectés (via WhatsApp) a également été testée.

Interactions Globales :

Tous les composants physiques ont été testés ensemble pour vérifier leur synchronisation et leur capacité à réagir de manière cohérente face à des variations simulées (par exemple, une baisse rapide de l'humidité du sol déclenche automatiquement la pompe).



Figure 30: simulation du système d'irrigation intelligent et de détection des maladies des plantes

Notifications Automatiques :

Les notifications ont été testées dans divers scénarios (par exemple, humidité du sol insuffisante ou détection de précipitations).

Les alertes automatiques ont été envoyées via WhatsApp pour informer l'utilisateur des besoins en irrigation ou des anomalies détectées.

La latence des notifications a été mesurée, avec une transmission moyenne inférieure à 5 secondes après la détection d'un événement critique.

Interface Utilisateur :

Le tableau de bord a été testé pour afficher en temps réel les données collectées (niveau d'humidité, état de la pompe, prévisions climatiques).

La visualisation graphique des données a permis aux utilisateurs d'interpréter facilement les informations et de surveiller le système.

Synchronisation Logicielle avec le Matériel :

La coordination entre les capteurs, les modèles d'analyse, et l'envoi de notifications a été validée dans des scénarios simulés pour garantir un flux d'informations sans interruptions.

Ces tests logiciels ont confirmé la capacité du système à répondre efficacement à différents cas d'usage, tout en offrant une interface fluide et des notifications en temps opportun pour améliorer l'expérience utilisateur.

FIN DU CHAPITRE IV

CONCLUSION

Les tests de simulation ont validé l'efficacité du système d'irrigation intelligent sur les plans matériel et logiciel. Les capteurs ont montré une grande fiabilité, et la coordination avec la pompe a été optimale. Les modèles SVM et LSTM ont permis des prédictions précises pour la gestion de l'irrigation. De plus, le modèle CNN a démontré sa capacité à détecter rapidement les maladies des plantes à partir des données visuelles simulées. Les notifications automatiques et l'interface utilisateur ont assuré une gestion réactive et intuitive. Ces résultats confirment la viabilité du système pour un déploiement terrain, offrant une solution complète pour une agriculture durable et efficace.



Conclusion et perspective

Le système proposé a permis d'introduire une approche novatrice pour l'irrigation intelligente et la gestion des maladies des plantes. En combinant l'utilisation de capteurs IoT, de modèles prédictifs avancés tels que les réseaux LSTM et CNN, et d'une interface utilisateur interactive, le projet a optimisé la consommation d'eau tout en offrant une solution proactive pour la surveillance de la santé des cultures. Concernant les maladies des plantes, le système a démontré une capacité remarquable à détecter précocement les anomalies, permettant ainsi des interventions ciblées et réduisant le recours excessif aux produits phytosanitaires. Cette double approche, axée sur la gestion durable des ressources hydriques et la protection phytosanitaire, a non seulement amélioré l'efficacité des pratiques agricoles, mais a également contribué à minimiser les pertes liées aux maladies, garantissant une meilleure productivité des cultures.

Pour l'avenir, plusieurs perspectives prometteuses se dessinent. Sur le plan technique, l'intégration de modèles d'apprentissage automatique encore plus performants, tels que des architectures hybrides combinant réseaux convolutifs profonds et méthodes traditionnelles, pourrait renforcer la précision des diagnostics. L'élargissement du système à des environnements agricoles variés et sa capacité à gérer simultanément plusieurs parcelles offriront une scalabilité accrue. L'ajout de technologies supplémentaires, telles que des drones équipés de capteurs spectraux et thermiques, pourrait améliorer la surveillance en temps réel des cultures et étendre la détection à un éventail plus large de maladies. Enfin, un accent particulier pourrait être mis sur l'évaluation de l'impact environnemental du système, en analysant la réduction des émissions de carbone et l'amélioration de l'efficience agricole, consolidant ainsi le rôle de cette solution dans la transition vers une agriculture durable et résiliente.

Références

1. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2018.01.009>.
2. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. arXiv preprint. <https://arxiv.org/abs/1606.05632>.
3. Picon, A., Alvarez-Gila, A., et al. (2019). Deep convolutional neural networks for plant disease detection. *Sensors*. <https://doi.org/10.3390/s19071561>.
4. Ramcharan, A., Baranowski, K., et al. (2019). Deep learning for image-based cassava disease detection. *PLOS ONE*. <https://doi.org/10.1371/journal.pone.0210796>.
5. Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
7. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*.
8. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*.
9. Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*.
10. Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017). Big Data in Smart Farming – A Review. *Agricultural Systems*.
11. Ray, D. K., Mueller, N. D., West, P. C., & Foley, J. A. (2013). Yield Trends Are Insufficient to Double Global Crop Production by 2050. *PLOS ONE*.
12. Documentation officielle TensorFlow/Keras : TensorFlow Guide.
13. Documentation Node-RED : Node-RED Guide.