

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences

Département d'informatique



MEMOIRE DE MASTER

En Informatique

Option : Ingénierie du Logiciel

THÈME :

**“Système IoT-ML avancé pour la
surveillance météorologique prédictive
et l’irrigation automatisée en
agriculture de précision”**

Réalisé par
Rezzoug Meriem
Sersoub Ikram

Encadré par
Dr. Lahiani Nesrine
Dr. Cheriguene Soraya

juin 2025

Remerciements

Avant toute chose, nous louons Allah, qui nous a accordé la force, la patience et la santé nécessaires pour mener à bien ce travail. Sans Sa grâce, aucun effort n'aurait abouti, et aucune ligne n'aurait été tracée.

Nous exprimons notre profonde gratitude à notre encadrante, ***Dr Nesrine Lahiani***, pour son encadrement rigoureux, sa générosité intellectuelle, sa disponibilité constante et ses conseils avisés tout au long de cette étude. Ses orientations précieuses ont constitué un repère fondamental dans notre cheminement.

Nos sincères remerciements vont également à notre co-encadrant, ***Dr Soraya Cheriguene***, pour sa bienveillance, son accompagnement attentif et ses remarques enrichissantes qui ont contribué à améliorer la qualité de notre travail.

Nous tenons à remercier les membres du jury qui nous font l'honneur d'évaluer ce mémoire. Leur temps, leur lecture attentive et leurs remarques constructives témoignent de leur engagement pour la formation et l'excellence académique.

Nos pensées les plus sincères vont à nos familles, véritables piliers de notre existence. À nos parents, pour leur amour, leurs prières, leurs sacrifices et leur soutien inconditionnel. À nos frères et sœurs, pour leur présence discrète, leurs encouragements et leur affection continue.

Enfin, nous adressons nos remerciements à toute personne ayant contribué, de près ou de loin, à la réalisation de ce mémoire. Que chacun trouve ici l'expression de notre profonde reconnaissance.

Qu'Allah vous bénisse tous et vous récompense pour votre bonté.

Dédicace

- À Toi, **ma mère** bien-aimée, Toi qui as bercé mes nuits d'inquiétude par des prières silencieuses, Toi dont l'amour inépuisable m'a donné la force de croire en moi, Ce travail est l'écho de ton sacrifice, de ta tendresse et de ta lumière. Qu'Allah te préserve et te comble de Ses bienfaits.
- À Toi, **mon père** respecté, Modèle de sagesse, de patience et d'intégrité, Toi qui m'as transmis la valeur de l'effort, du savoir et de la dignité, Ce mémoire est un fruit que je dépose avec respect entre tes mains. Qu'Allah te récompense pour chaque pas fait en silence pour moi.
- À mes frères **Youcef** , **Ayoub** et **Idris** , Merci d'avoir été ces piliers discrets mais solides, Présents dans l'ombre, avec des mots simples mais puissants, Vous êtes la chaleur de ma maison et la force de mon dos.
- À ma famille élargie, Tantes, oncles, cousins, cousines... **Hadjer Maroua Ahlem Boutaina Hala Houda** Votre affection et vos encouragements m'ont toujours porté plus loin que je ne l'imaginais.
- Et à vous, mes chers amis **Rania** , **Asmaa** , **Yousra** et **Loubna** Compagnons de route, rires partagés, défis relevés, Merci pour votre présence, vos mots sincères, votre soutien dans les moments d'épuisement et de doute. Vous avez embelli ce chemin.

Meriem

À ma chère **maman**,

Ton amour inépuisable, ta patience et tes prières m'ont porté tout au long de ce parcours. Tu es mon pilier et ma source d'inspiration.

À mon cher **papa**,

Merci pour ta sagesse, ton soutien silencieux mais constant, et pour m'avoir toujours poussé à donner le meilleur de moi-même.

À ma précieuse **sœur Manel**,

Ta tendresse, ton écoute et ta présence m'ont souvent réconforté dans les moments difficiles.
Merci d'être toujours là.

À mon frère **Ahmed**,

Ton humour, ta force et ta complicité m'ont aidé à garder le sourire même dans les moments de doute. Merci d'être un frère sur qui je peux toujours compter.

Ikram

0.1 Résumé

L'agriculture est un élément crucial pour le développement économique de chaque nation. Face à la croissance démographique, aux fluctuations climatiques constantes et aux ressources restreintes, satisfaire les exigences nutritionnelles de la population contemporaine s'avère de plus en plus complexe.

L'agriculture de précision, aussi appelée agriculture intelligente, s'est imposée comme un instrument novateur pour répondre aux enjeux contemporains de la durabilité dans le secteur agricole. Le mécanisme qui anime cette technologie de pointe est l'apprentissage automatique (ML). Cela permet à la machine d'acquérir des connaissances sans être spécifiquement codée. L'intelligence artificielle et les équipements agricoles compatibles IoT (Internet des objets) constituent des facteurs essentiels de la prochaine révolution dans le domaine agricole.

La technologie agricole intelligente basée sur les technologies IoT présente de nombreux avantages dans tous les processus et pratiques agricoles en temps réel, notamment l'irrigation, la protection des plantes, l'amélioration de la qualité des produits .

Notre projet vise à répondre à ce besoin crucial en développant un Système IoT-ML avancé pour la surveillance météorologique prédictive et l'irrigation automatisée en agriculture de précision en collectant et transmettant les données agricoles en temps réel, à l'aide d'algorithmes ML pour rendre les pratiques agricoles plus contrôlées et optimisées. Nous avons créé des modèles de prévision avancés pour anticiper la météo et les précipitations en fonction des données obtenues à partir de capteurs et optimiser en conséquence le système d'irrigation avec un système d'aide à la décision .

Mots clés :

apprentissage automatique (ML) , irrigation, L'intelligence artificielle, Internet des objets (IoT), L'agriculture de précision , Surveillance , aide à la décision .

0.2 Abstract

Agriculture plays a vital role in the economic growth of any country. With the increase of population, frequent changes in climatic conditions and limited resources, it becomes a challenging task to fulfil the food requirement of the present population.

Precision agriculture also known as smart farming have emerged as an innovative tool to address current challenges in agricultural sustainability. The mechanism that drives this cutting edge technology is machine learning (ML). It gives the machine ability to learn without being explicitly programmed. ML together with IoT (Internet of Things) enabled farm machinery are key components of the next agriculture revolution.

Smart farming technology based on IoT technologies has many benefits in all real-time farming processes and practices including irrigation, plant protection, improving product quality.

Our project aims to meet this crucial need by developing a System based on Advanced IoT-ML System for Predictive Weather Monitoring and Automated Irrigation in Precision Agriculture by collecting and transmitting agricultural data in real time, using ML algorithms to make agricultural practices more controlled and optimized. We have created advanced forecasting models to anticipate weather and precipitation based on data obtained from sensors and optimize the irrigation system accordingly using a decision support system.

Keywords :

machine learning (ML), irrigation, artificial intelligence, internet of things (IoT), precision agriculture, decision support , monitoring.

الملخص

تُعدّ الزراعة عنصرًا حيويًا في التنمية الاقتصادية لكل دولة. ومع النمو السكاني، والتغيرات المناخية المستمرة، وندرة الموارد، أصبح من الصعب بشكل متزايد تلبية الاحتياجات الغذائية للسكان المعاصرين

لقد فرضت الزراعة الدقيقة، المعروفة أيضًا باسم الزراعة الذكية، نفسها كأداة مبتكرة لمواجهة التحديات المعاصرة ، والذي يتيح للألة (ML) المتعلقة بالاستدامة في القطاع الزراعي. وتستند هذه التكنولوجيا المتقدمة إلى التعلم الآلي اكتساب المعرفة دون الحاجة إلى برمجتها بشكل صريح. تُعدّ الذكاء الاصطناعي والمعدات الزراعية المتوافقة مع إنترنت من العوامل الأساسية في الثورة الزراعية القادمة (IoT) الأشياء

توفر التكنولوجيا الزراعية الذكية، القائمة على تقنيات إنترنت الأشياء، العديد من المزايا في جميع العمليات والممارسات الزراعية في الوقت الفعلي، بما في ذلك الري، وحماية النباتات، وتحسين جودة المنتجات

لمراقبة الطقس ML و IoT يهدف مشروعنا إلى تلبية هذه الحاجة الملحة من خلال تطوير نظام متقدم يعتمد على تقنيات بشكل تنبؤي وللاي في إطار الزراعة الدقيقة. ويتم ذلك عبر جمع ونقل البيانات الزراعية في الوقت الفعلي، واستخدام خوارزميات التعلم الآلي لجعل الممارسات الزراعية أكثر تحكمًا وتحسينًا. لقد أنشأنا نماذج تنبؤ متقدمة للتنبؤ بالطقس وكميات الأمطار، استنادًا إلى البيانات التي يتم الحصول عليها من أجهزة الاستشعار، ومن ثم تحسين نظام الري وفقًا لذلك عبر نظام دعم اتخاذ القرار

:الكلمات المفتاحية

الزراعة الدقيقة، المراقبة، دعم اتخاذ القرار (IoT) ، الري، الذكاء الاصطناعي، إنترنت الأشياء (ML)، التعلم الآلي .

Liste des Acronymes

CNN Convolutional Neural Network

CSS Cascading Style Sheets

DL Deep Learning .

HTML HyperText Markup Language

IA Intelligence Artificielle.

IDE Integrated Development Environment

IoT Internet des Objets.

IP Internet Protocol.

IQR Interquartile Range

ITGC institut technique des grandes cultures

JAS Jours Après Semis

LSTM Long Short-Term Memory

MAE Mean Absolute Error

ML Machine Learning.

MSE Mean Squared Error

NoSQL Not Only SQL.

QR Quick Response.

RFID Radio Frequency Identification.

RNN Recurrent Neural Network.

SQL Structured Query Language

SVM Support Vector Machine.

URI Uniform Resource Identifier.

Wi-Fi Wireless Fidelity.

WNN Wavelet Neural Network

Table des matières

0.1	Résumé	II
0.2	Abstract	III
Table des figures		X
Liste des tableaux		XII
Introduction Générale		2
	Contexte de travail	2
	Problématique	2
	Objectifs du travail	3
	Organisation du mémoire	3
1	État de l’art	5
1.1	Introduction	5
1.2	Agriculture et Intelligence Artificielle	5
1.2.1	Application de l’IA dans l’agriculture	6
1.2.2	L’impact de l’IA dans l’agriculture	8
1.2.3	Techniques d’apprentissage automatique pour l’analyse des données agricoles	8
1.3	Internet des Objets	10
1.3.1	Définition	11
1.3.2	Domaines d’application de l’IoT	11
1.3.3	Processus de fonctionnement d’un d’IoT	13
1.3.4	Architecture d’un système d’IoT	14
1.3.5	Avantages et inconvénients de l’Iot :	15
1.3.5.1	Avantages :	15
1.3.5.2	inconvénients	15
1.3.6	Capteurs de surveillance météorologique	16
1.4	Comparaison des travaux existants	17

1.4.1	prévision météorologique	17
1.4.1.1	Travail de : Yashon O. Ouma et al.	17
1.4.1.2	Travail de : Nitin Singh et al.	17
1.4.1.3	Travail de : Pan, B., Hsu, K., AghaKouchak, A., & Sorooshian, S.	18
1.4.2	prévision d'irrigation	19
1.4.2.1	Travail de : Youness Tace, Mohamed Tabaa, Sanaa Elfilali, Cherkaoui Leghris, Hassna Bensag, Eric Renault	19
1.5	conclusion	20
2	Conception du système	21
2.1	Introduction	21
2.2	Motivation de l'étude de cas	21
2.3	Processus de conception du système	22
2.4	Prévision météorologique	23
2.4.1	Dataset	23
2.4.2	Prétraitement des Données	24
2.4.2.1	Enrichissement des données	25
2.4.2.2	Normalisation des données	25
2.4.2.3	Nettoyage des données	25
2.4.3	Construction du Modèle de Prévision Météorologique	26
2.4.3.1	Random forest	26
2.4.3.2	Support Vector Regressor (SVR)	27
2.4.3.3	LSTM simple(Long Short-Term Memory)	28
2.4.3.4	LSTM Profond	28
2.4.4	Evaluation des Modèles et Sélection du Meilleur	30
2.5	Système d'Irrigation	31
2.5.1	Paramètres d'entrée du système	31
2.5.2	Algorithme de déclenchement de l'irrigation	32
2.5.3	Sorties du système de décision	35
2.6	Conclusion	36
3	Implémentation et test du système	37
3.1	Introduction	37
3.2	Environnement de développement	37
3.2.1	Outils de programmation	37
3.2.2	Back-end	38
3.2.3	Front-end	38
3.2.4	Bibliothèques utilisées	38
3.3	Cas d'étude	39

3.3.1	Prétraitement des Données	40
3.3.1.1	Enrichissement des données	40
3.3.1.2	Nettoyage des données	41
3.3.2	Construction du Modèle de Prévion des Précipitations	42
3.3.3	Les interfaces	43
3.3.3.1	Page de Surveillance météorologique et précipitations	43
3.3.3.2	Formulaire de données climatiques et du sol	44
3.3.3.3	Page de Simulation intelligente d'irrigation	45
3.3.4	Validation du système	46
3.3.4.1	La validation en consultant l'expert du domaine.	46
3.3.4.2	La validation par les mesures de performance	47
3.3.5	Discussion	49
3.3.5.1	Analyse des résultats obtenus	49
3.3.5.2	Comparaison des approches algorithmiques pour la prévision des précipitations	50
3.3.5.3	Tests expérimentaux et choix de l'architecture LSTM	51
3.3.6	Conclusion	53
	Conclusion et perspectives	54
	Bibliographie	56

Table des figures

1.1	Application de l'IA dans l'agriculture	7
1.2	Techniques d'apprentissage automatique appliquées à l'agriculture	8
1.3	Domaine d'application iot [23]	13
1.4	Architecture de l'IOT	14
2.1	Processus de conception du système	22
2.2	Architecture de LSTM profond	29
2.3	les stades de croissance du blé.	32
3.1	Pseudo-code pour le Regroupement des données par jour	40
3.2	Pseudo-code pour Convertir la colonne 'date' au bon format	40
3.3	Pseudo-code pour fusionner les deux jeux de données	40
3.4	Pseudo-code pour sélectionner les variable pertinente et ignoré les inutiles	41
3.5	Pseudo-code pour la transformation logarithmique log1p	41
3.6	Pseudo-code pour le traitement des valeurs aberrantes	41
3.7	Pseudo-code pour détecter les valeurs manquantes	42
3.8	Pseudo-code du Modèle de Prédiction des Précipitations	43
3.9	Pseudo-code pour l'Entraînement et Sauvegarde du Modèle LSTM avec Régulation Automatique	43
3.10	les conditions météorologique captées	44
3.11	les prévisions de précipitations sur 7 jours	44
3.12	Graphique des précipitations prévues	44
3.13	formulaire utilisateur pour les données climatiques et du sol	45
3.14	Simulation automatique de l'irrigation	45
3.15	Résultat d'irrigation sur 7 jours	46
3.16	Analyse comparative des performances des modèles selon les métriques MAE, MSE et R^2	49
3.17	pseudo-code pour le premier test du modèle LSTM.	51
3.18	pseudo-code pour le deuxieme test du modèle LSTM.	51

3.19	pseudo-code pour le troisiemme test du modèle LSTM.	52
3.20	pseudo-code pour le quatriemme test du modèle LSTM.	52

Liste des tableaux

1.1	Tableau comparatif des performances des modèles de Prévion des précipitations .	10
1.2	Exemple d'application de l'IoT	11
1.3	processus de fonctionnement de l'IOT [24]	14
1.4	Capteurs utilisés pour la surveillance environnementale et agricole	17
1.5	Résumé du travail sur la prévision météorologique utilisant l'apprentissage automatique	19
1.6	Résumé du travail sur la prévision d'irrigation utilisant l'apprentissage automatique	20
2.1	Description des attributs du jeu de données météorologiques	24
2.2	Consommation d'eau journalière moyenne selon les stades du blé[46]	31
2.3	Correspondance entre stade phénologique du blé et jours après semis[47]	32
2.4	Dose d'irrigation recommandée selon le stade phénologique[46].	36
2.5	Fréquence d'irrigation selon le type de sol et sa capacité de rétention [46].	36
3.1	Tableau de validation du système d'irrigation par comparaison aux décisions de l'expert	47
3.2	Tableau comparatif des performances des modèles de Prévion des précipitations .	49
3.3	Tableau comparatif des approches prédictives	50

Introduction Générale

Contexte de travail

L'agriculture de précision transforme la façon dont les agriculteurs gèrent leurs cultures, en s'appuyant sur des technologies avancées pour améliorer l'efficacité et la productivité. L'intégration de l'Internet des objets (IoT) a permis de surveiller en temps réel des facteurs environnementaux tels que l'humidité, la température et l'humidité du sol. Dans le même temps, les techniques d'apprentissage automatique révolutionnent l'analyse prédictive, permettant des prévisions météorologiques précises et la reconnaissance de modèles. En combinant ces technologies, les agriculteurs peuvent automatiser les systèmes d'irrigation, optimiser l'utilisation des ressources et prendre des décisions éclairées basées sur des informations basées sur des données. Cette évolution vers une agriculture intelligente et intensive en données recèle un immense potentiel d'amélioration des rendements et de promotion de la durabilité [1].

Problématique

Malgré la disponibilité des technologies modernes, de nombreux agriculteurs sont encore confrontés à des défis importants dans la gestion des ressources en eau et l'adaptation aux conditions météorologiques imprévisibles. Les systèmes d'irrigation traditionnels conduisent souvent à une utilisation inefficace de l'eau, soit par irrigation excessive, soit par irrigation insuffisante, ce qui peut avoir un impact négatif sur la santé des cultures et gaspiller des ressources. De plus, les conditions météorologiques sont un facteur critique dans l'agriculture, et des prévisions inexacts ou retardées peuvent entraîner une mauvaise planification et des dommages aux cultures. Les solutions existantes ne permettent pas d'intégrer la surveillance environnementale en temps réel avec des prévisions météorologiques précises et un contrôle automatisé de l'irrigation. Il est nécessaire de disposer d'un système intelligent capable de prédire les conditions météorologiques, de surveiller la santé des cultures et d'automatiser l'irrigation en fonction de données en temps réel et de prévisions futures pour améliorer la gestion et la durabilité des exploitations agricoles.

Objectifs du travail

Ce projet vise à développer un système avancé qui intègre la technologie IoT avec des modèles d'apprentissage automatique basés sur LSTM pour améliorer la surveillance météorologique et automatiser les processus d'irrigation en utilisant un système d'aide à la décision pour l'agriculture de précision. Les principaux objectifs sont les suivants :

1. Surveillance environnementale en temps réel :
 - Déployer des capteurs IoT dans toute la ferme pour surveiller en permanence les facteurs environnementaux critiques tels que l'humidité du sol, la température, l'humidité et l'intensité lumineuse.
 - Collecter et transmettre des données en temps réel au système central pour analyse.
2. Surveillance météorologique prédictive :
 - Mettre en œuvre un modèle d'apprentissage automatique pour analyser les données météorologiques historiques et prédire les conditions météorologiques futures telles que les précipitations, les variations de température et l'humidité.
 - Fournir des prévisions météorologiques précises adaptées à la région spécifique de l'exploitation, permettant aux agriculteurs de prendre des décisions opportunes.
3. Système d'irrigation automatisé :
 - Développer un système de contrôle d'irrigation intelligent qui utilise les données d'humidité du sol en temps réel et les prévisions météorologiques pour automatiser la distribution de l'eau.
 - Optimiser l'irrigation en ajustant le débit d'eau en fonction des prévisions météorologiques, des conditions du sol et des besoins des cultures, garantissant ainsi une utilisation efficace de l'eau.
4. Optimisation de la santé et de la croissance des cultures :
 - Utilisez les données environnementales et les prévisions météorologiques pour fournir des informations exploitables aux agriculteurs, les aidant ainsi à optimiser les calendriers de plantation, l'allocation des ressources et les stratégies de lutte antiparasitaire.
 - Réduisez le stress des cultures et améliorez les conditions de croissance en automatisant l'irrigation sur la base de données précises.

Organisation du mémoire

Ce travail se structure en trois sections majeures, chacune apportant une contribution à l'élaboration progressive de notre système d'irrigation intelligent :

Chapitre 1 : État de l'art

Ce chapitre offre un aperçu global des concepts et technologies qui constituent la base de notre projet. Cela débute par une étude détaillée de l'application de IA dans le secteur agricole,

son influence sur la productivité et la gestion des ressources, ainsi que les principales méthodes d'apprentissage automatique employées pour l'analyse des données agricoles, on passe par la suite à technologie de IOT, nous abordons ses principaux domaines d'application, l'architecture typique des systèmes IoT, ainsi que leurs avantages et inconvénients et les capteur utilisé pour la prévision météorologique et l'irrigation, le chapitre traite aussi une étude comparative de diverses recherches actuelles.

Chapitre 2 : Conception du système

Dans ce segment, nous expliquons en détail le processus de création du système intelligent. Il commence par exposer les motivation de l'étude de cas, avant d'expliquer l'ensemble du processus de conception.

Ce chapitre est structuré autour de deux axes principaux :

- Prévisions météorologique : Elle explique l'utilisation des jeux de données, leur combinaison pour compléter les colonnes absentes, ainsi que les phases de prétraitement des données (enrichissement, normalisation, nettoyage, identification des valeurs extrêmes). Par la suite, nous effectuons une comparaison entre divers modèles ML DL, dans le but de choisir celui qui présente les meilleures performances prédictives.

- Système d'aide à la décision pour l'irrigation :

cette section détaille l'application des normes agronomiques et des critères provenant des documents de référence (ITGC et Zadoks) pour élaborer un système d'irrigation intelligent. On y établit les critères de prise de décision, la fréquence d'arrosage en fonction du type de sol, l'évaluation des phases de croissance du blé, et comment les précipitations attendues peuvent décider si l'irrigation doit être mise en place ou non.

Chapitre 3 : Implémentation et test du système

Le dernier chapitre présente l'application concrète du système élaboré. Nous expliquons l'environnement de développement, ainsi que les outils et les langages employés pour la conception de notre application web. Nous mettons aussi en avant les principales interfaces de l'application. Pour finir, on procède à une évaluation du système en effectuant des tests de performance. Ces évaluations servent à juger l'efficacité de notre modèle prédictif, ainsi que la cohérence et l'adéquation des choix d'irrigation suggérés par le système d'aide à la décision.

Chapitre 1

État de l'art

1.1 Introduction

Avec la popularité grandissante de l'intelligence artificielle et l'internet des objets (IoT), les dispositifs connectés ont envahi tous les domaines de notre existence, qu'il s'agisse de la santé, de la maison intelligente, du secteur automobile ou encore des villes intelligentes. Il est donc cohérent de voir l'IoT et l'IA s'appliquer dans le domaine agricole.

Ce chapitre offre un aperçu de l'état actuel des connaissances en étudiant l'agriculture et son passage à une vision plus technologique grâce à l'incorporation de l'IoT et du Machine Learning dans l'agriculture de précision. Par la suite, nous examinerons les principes de base de l'IoT, ses champs d'utilisation et son principe de fonctionnement.

Nous mettrons en avant ses avantages, ses obstacles. Enfin nous abordons les capteurs et les technologies utilisées pour automatiser l'irrigation et améliorer l'efficacité des systèmes agricoles.

Ce chapitre conclura par une comparaison avec les travaux existants, nous permettant d'évaluer les approches développées jusqu'à présent et d'identifier les domaines à améliorer pour notre étude.

1.2 Agriculture et Intelligence Artificielle

Avec l'augmentation de la population au cours de l'année, la demande en produits agricoles augmente de jour en jour. Cependant, la disponibilité limitée des terres agricoles et la diminution de l'intérêt des jeunes générations pour l'agriculture comme profession rendent difficile pour l'industrie agricole de satisfaire les besoins alimentaires de millions de personnes. Aujourd'hui, l'industrie agricole adopte largement des technologies intelligentes comme l'IoT et l'IA pour cultiver efficacement des produits biologiques sur des superficies limitées et pour surmonter les défis traditionnels des agriculteurs [2].

L'IA est l'une des technologies qui connaît la croissance la plus rapide dans ce monde moderne. L'IA est désormais largement utilisée dans diverses industries allant du développement

de la robotique à la surveillance des problèmes environnementaux. L'automatisation médiée par l'IA de plusieurs secteurs, y compris l'agriculture, générera des produits de bonne qualité avec moins d'impacts négatifs sur l'environnement. L'IA utilise une variété de capteurs intelligents qui traitent les données en temps réel avec une plus grande précision. Par conséquent, l'IA a amélioré les performances de tout appareil à plusieurs reprises avec une plus grande sensibilité et précision dans leurs résultats [3].

L'agriculture intelligente, qui utilise l'intelligence artificielle, favorise l'agriculture de précision grâce à des technologies modernes et sophistiquées. Elle permet aux agriculteurs de surveiller les plantes à distance et facilite les processus agricoles, comme la récolte et l'amélioration du rendement des cultures.

Grâce à l'automatisation des capteurs et des machines, la main-d'œuvre agricole devient plus efficace. Cette approche offre de nombreux avantages en temps réel dans l'ensemble des procédures agricoles, notamment la gestion de l'irrigation, la protection des cultures, l'optimisation de la qualité des récoltes, la fertilisation, ainsi que l'anticipation des maladies. [4].

1.2.1 Application de l'IA dans l'agriculture

Comme dans beaucoup de secteurs, l'agriculture moderne fait très largement appel à des solutions numériques. L'IA est bien entendu très présente. Dans cette partie, nous présentons les principaux domaines d'application de l'IA dans le secteur agricole, comme illustré dans la Figure 1.1 :

Propriétés du sol et prévisions météorologiques : La teneur en humidité du sol est un paramètre important à prendre en compte dans l'industrie agricole car elle permet une planification précise de l'irrigation.

La prévision des précipitations joue un rôle essentiel dans le système de gestion des ressources en eau, Il est très difficile pour les approches statistiques de prédire avec précision les précipitations [2].

Systèmes de surveillance de la santé des cultures : Les techniques de télédétection, l'imagerie hyperspectrale et l'IA permettent de créer des systèmes de surveillance de la santé des cultures capables de surveiller la santé des cultures en termes de temps et d'efforts. L'IA facilite l'identification des problèmes de ravageurs, de maladies et de mauvaises herbes et automatise la gestion de ces problèmes. Les systèmes de production agricole basés sur l'IA prédisent également les situations futures et émettent des avis pour les semis, la lutte contre les ravageurs et la tarification des produits de base [5].

Systèmes d'irrigation intelligents : l'irrigation intelligente consiste à fournir la bonne quantité d'eau au bon endroit et au bon moment pour la bonne culture afin d'améliorer le rendement des cultures. Grâce à un système d'irrigation automatisé basé sur des capteurs, les problèmes liés à la faible efficacité de l'irrigation peuvent être résolus dans une plus large mesure. De plus, la planification automatisée de l'irrigation est possible à l'aide de machines entraînées

sur les conditions météorologiques historiques et la qualité du sol de la localité ainsi que sur le type de cultures à cultiver [5].

Production et gestion du bétail : L'intelligence artificielle, l'IoT et les technologies Blockchain sont largement explorées pour améliorer la durabilité du bétail et pour analyser leurs habitudes de station debout, de déplacement, de boisson et d'alimentation, indiquer le niveau de stress que subit l'animal, ce qui permet de prédire la vulnérabilité aux maladies, la prise de poids et la production du bétail. Le bien-être animal concerne généralement la santé de l'animal ; pour cela, les techniques d'apprentissage automatique sont appliquées à leur fonction de surveillance de la santé pour une détection prospective des maladies . Alors que la production animale utilise l'apprentissage automatique pour estimer la production équilibrée de bétail afin que les producteurs obtiennent des avantages économiques. [2].

Techniques de récolte intelligentes : Les systèmes de récolte intelligents aident les agriculteurs à récolter les produits agricoles en réduisant les efforts humains. Des technologies telles que les capteurs intelligents, la robotique, les drones et l'IA et les techniques de vision par ordinateur basées sur le ML sont utilisées pour récolter intelligemment les cultures. La récolte intelligente offre une meilleure connaissance des cultures et aide les agriculteurs à atteindre le potentiel de récolte des cultures, ce qui conduit à une productivité accrue [2].

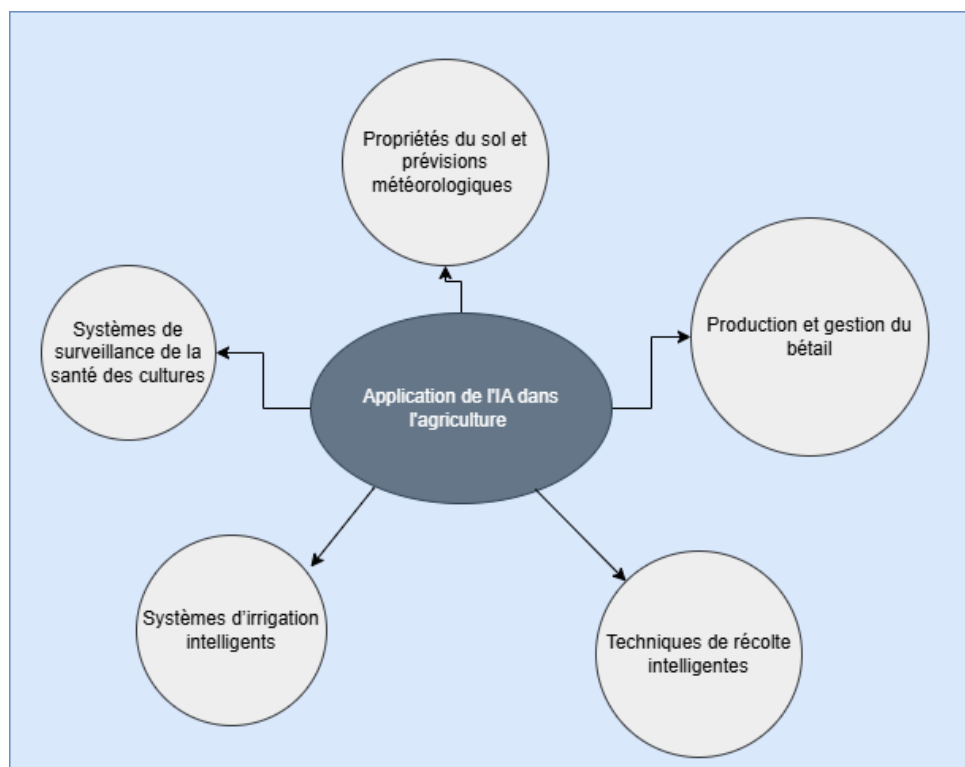


FIGURE 1.1 – Application de l'IA dans l'agriculture

1.2.2 L'impact de l'IA dans l'agriculture

L'agriculture est un secteur essentiel nécessaire à la survie de la communauté humaine. Plusieurs mesures ont été prises pour améliorer la production agricole. Cependant, les conditions environnementales difficiles et les infestations fréquentes de parasites entraînent des pertes agricoles. Dans un tel scénario, l'intégration de technologies avancées telles que l'Intelligence Artificielle (IA) et des capteurs avancés couplés à l'Internet des objets (IoT) pourrait augmenter la production agricole et minimiser les pertes économiques [3].

Les drones révolutionnent le secteur agricole. Ces drones sont équipés de caméras et sont utilisés pour différentes applications telles que la surveillance des champs et des cultures, la pulvérisation de pesticides et l'irrigation goutte à goutte. Les images capturées par les drones tout au long du cycle de vie des cultures peuvent être examinées à l'aide d'algorithmes de DL et de vision par ordinateur pour l'identification des maladies et des mauvaises herbes.

Les applications d'assistant virtuel basées sur le traitement du langage naturel (NLP), comme les chatbots, peuvent informer les agriculteurs des dernières avancées technologiques dans le domaine de l'agriculture. Les agriculteurs peuvent trouver des solutions à leurs problèmes et intégrer les dernières technologies dans leur exploitation pour améliorer la productivité de leurs champs. l'IA constitue ainsi une technologie clé qui joue un rôle essentiel dans le secteur agricole [2].

1.2.3 Techniques d'apprentissage automatique pour l'analyse des données agricoles

Les techniques d'apprentissage automatique jouent un rôle clé dans l'analyse des données agricole, en optimisant la prise de décision et la gestion des ressources. La FIGURE 1.2 illustre les principales méthodes utilisées et leur applications dans différents domaines agricoles.

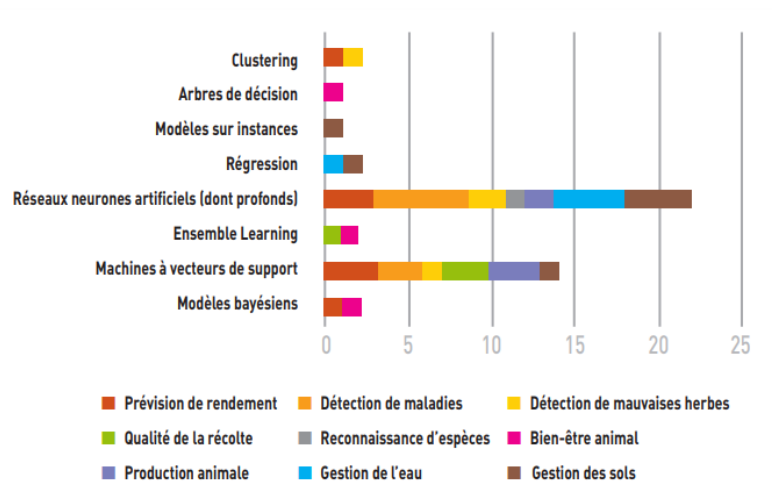


FIGURE 1.2 – Techniques d'apprentissage automatique appliquées à l'agriculture

Le tableau ci-dessous présente les principales techniques d'apprentissage automatique, leurs définitions et leurs applications spécifiques dans le domaine agricole. Il met en lumière comment ces méthodes optimisent les pratiques et améliorent les prises de décision en agriculture.

Technique	Définition	Application dans des données agricoles
Régression	La régression en machine learning est une technique statistique utilisée pour modéliser et analyser la relation entre une variable cible et une ou plusieurs variables explicatives. [7] - Régression linéaire : suppose une relation droite entre la variable cible et les variables explicatives. [7] - Régression multiple : prend en compte plusieurs variables explicatives pour modéliser des situations complexes. [7]	- Modèles de prévision des rendements : analyse les données historiques (température, précipitations, fertilité des sols) pour prédire les rendements futurs. [8] - Optimisation de l'utilisation des engrais : détermine le niveau optimal d'application d'engrais pour maximiser le rendement des cultures. [8]
Classification traditionnelle	Méthode d'apprentissage supervisée où le modèle prédit l'étiquette correcte d'une donnée d'entrée. [9] Algorithmes courants : - Machine à vecteurs de support (SVM)[10] - Arbres de décision [10] - Répartition en K-moyennes (K-means) [10]	- Classification des cultures par télédétection : attribue une classe à chaque type de culture en combinant les données radar et l'imagerie optique. [11] - Classement et tri des produits : utilise la vision par ordinateur pour trier les produits agricoles en fonction de leur qualité (taille, forme, couleur, volume)[12].
Clustering	Technique non supervisée regroupant des exemples non étiquetés selon leur similarité [13]. Types : - Clustering basé sur centroïde - Clustering basé sur la densité - Clustering basé sur la distribution - Clustering hiérarchique	- Délimitation des zones de gestion : identifie des zones de potentiel de rendement basé sur des cartes historiques et les propriétés du sol. [14] - Surveillance des mauvaises herbes : reconnaissance des mauvaises herbes sans connaissances préalables sur les espèces.[15]

Réseaux de neurones artificiels (RNA)	Modèle inspiré du cerveau humain, constitué de couches de neurones artificiels (entrée, cachées, sortie) pour prendre des décisions complexes. [16]	- Prédiction du rendement des cultures : analyse les résultats précédents et les facteurs environnementaux pour estimer les rendements futurs. [17] - Surveillance de la santé des plantes : utilise la vision par ordinateur pour suivre les stades de croissance et détecter les anomalies. [12]
Apprentissage profond (Deep Learning)	Sous-catégorie du machine learning utilisant des réseaux neuronaux multicouches pour des tâches complexes. [18] Types : [18] - Réseaux neuronaux convolutifs (CNN) - Réseaux neuronaux récurrents (RNN) - Auto-encodeurs variationnels (VAE) - Réseaux antagonistes génératifs (GAN)	- Détection/reconnaissance de plantes et fruits : identifie les maladies, insectes et dégâts foliaires via l'imagerie. [19] - Estimation des récoltes et rendements : utilise des CNN pour estimer les rendements à partir d'images à haute résolution. [19]

TABLE 1.1 – Tableau comparatif des performances des modèles de Prédiction des précipitations

1.3 Internet des Objets

L'un des mots à la mode dans le domaine des technologies de l'information est l'Internet des objets (IoT). L'avenir est l'Internet des Objets, qui transformera les objets du monde réel en objets virtuels intelligents. L'IoT vise à unifier tout ce qui existe dans notre monde sous une infrastructure commune, nous donnant non seulement le contrôle des choses qui nous entourent, mais nous tenant également informés de l'état des choses.

Au début des années 1980, le premier appareil connecté à Internet était un distributeur automatique de Coca-Cola à l'Université Carnegie Mellon. Même si le terme « Internet des objets » à la mode date de plusieurs années en référence à une machine à café, l'expression initiale a vu le jour grâce à Kevin Austin, qui était alors directeur exécutif des laboratoires Auto-ID du MIT, en 1999. L'idée de l'IoT a gagné en popularité grâce au centre Auto-ID en

2003 ainsi qu'aux diverses publications d'analystes de marché associés [20].

1.3.1 Définition

Chaque fois que vous regardez votre montre intelligente pour compter le nombre de calories dépensées au cours de votre jogging ou demandez à Alexa ou Siri de calculer la valeur énergétique d'une pâtisserie, vous utilisez la technologie IoT [21]. La meilleure définition de l'Internet des objets serait : « Un système interconnecté et exhaustif d'entités numériques intelligentes qui possèdent la capacité de s'organiser de manière autonome, d'échanger des informations ainsi que des données et des ressources, et de réagir et d'agir en réponse aux situations et aux variations de l'environnement » [20].

1.3.2 Domaines d'application de l'IoT

Le marché mondial de l'IoT est vaste et offre un large potentiel d'applications dans tous les secteurs. Ces nouvelles applications révolutionnent les usages et impactent le quotidien des individus, des entreprises et des territoires. Voici quelques secteurs dans lesquels l'Internet des objets a eu un impact significatif :

TABLE 1.2 – Exemple d'application de l'IoT

Domaine	Explication
Fabrication/Industrie	De nombreux projets d'automatisation et de contrôle d'usine basés sur l'IoT incluent des solutions complètes d'usine intelligente telles que la surveillance de l'atelier de production, des automates programmables à distance ou des systèmes automatisés de contrôle qualité. Les projets hors usine typiques incluent le contrôle à distance des machines connectées et la surveillance des équipements. [22]
Transport	Maintien de la santé du véhicule : la technologie de maintenance prédictive repose sur l'utilisation d'outils de communication IoT qui collectent des données sur les performances de différentes pièces, et évaluent le risque d'un dysfonctionnement du matériel ou logiciel du véhicule. [22]

Santé	La demande pour des applications IoT spécifiques à la santé, telles que la consultation en télésanté, le diagnostic numérique, la surveillance à distance et l'assistance robotique, augmente. Elles aident à réduire le temps d'attente aux urgences, suivre les patients et les stocks, améliorer la gestion des médicaments, et assurer la disponibilité des appareils médicaux. [22]
Villes intelligentes	Grâce à la puissance de l'IoT, des villes entières deviennent interconnectées. En analysant les données issues de capteurs urbains, les villes peuvent améliorer les infrastructures, la sécurité, le transport et les services publics. [22]
Énergie	L'IoT est utilisé pour surveiller les performances des systèmes énergétiques. Il permet d'optimiser l'efficacité des centrales, y compris à charbon, et de réduire les émissions. [22]
Vente au détail	Les détaillants utilisent l'IoT pour améliorer l'expérience client avec l'affichage numérique, le suivi des clients, le contrôle des marchandises, et la gestion intelligente des stocks. [22]
Agriculture	L'IoT permet la collecte de données à distance, le contrôle des effets climatiques, et l'optimisation des ressources pour améliorer le rendement des cultures. [4]
Chaîne d'approvisionnement	Les dispositifs IoT permettent le suivi GPS des produits, la surveillance des conditions de stockage, et l'authentification des expéditions, améliorant ainsi la qualité et la traçabilité. [22]

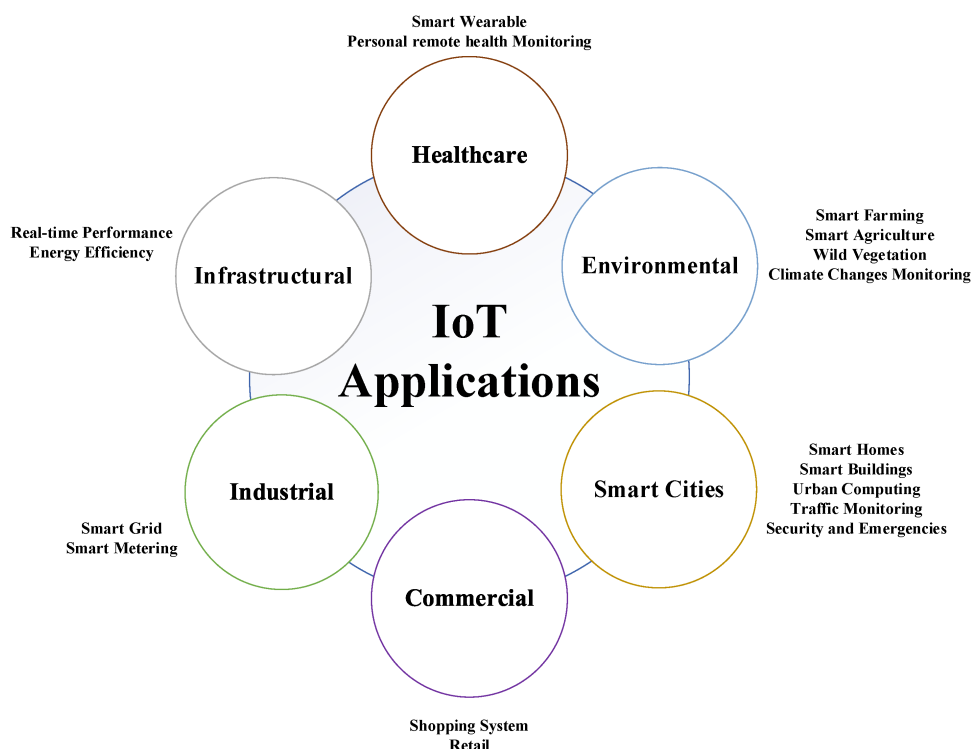


FIGURE 1.3 – Domaine d'application iot [23]

1.3.3 Processus de fonctionnement d'un d'IoT

le processus de fonctionnement de l'IOT repose sur l'interconnexion d'appareils intelligents , facilitant la collecte , la transmission et l'analyse des données . le tableau suivant détaille les étapes clés de ce processus.

Étape	Explication
Identification	Les composants connectés dans un réseau IoT doivent être adressables et identifiables pour assurer leur bon fonctionnement. Différents identifiants peuvent être utilisés, tels que les URI (Uniform Resource Identifiers) et les adresses IP (Internet Protocol).
Détection	Les systèmes IoT sont équipés de capteurs qui collectent des données provenant de leur environnement. Ces capteurs peuvent mesurer différentes caractéristiques telles que la température, la position, la proximité...
Communication	Une fois les données collectées par les capteurs, elles doivent être transmises à d'autres systèmes capables de les analyser et de prendre des décisions en fonction des résultats. Différentes technologies de communication sont utilisées, telles que le Wi-Fi, le Bluetooth, les codes QR et les étiquettes RFID

Informatisation	Les données collectées sont traitées par des microcontrôleurs ou d'autres dispositifs similaires pour permettre leur traitement par le système IoT. Par exemple, les détecteurs de fumée peuvent automatiquement appeler les pompiers en cas d'incendie.
Services	Les systèmes IoT offrent différents services et fonctionnalités en fonction de leur application spécifique. Par exemple, un système de maison intelligente peut contrôler l'éclairage d'une pièce à l'aide d'actionneurs.
Sémantique	La sémantique joue un rôle important dans l'IoT en fournissant un modèle commun de données qui peut être réutilisé et partagé entre les applications et les systèmes. Cela facilite le tri et l'interprétation des données de manière intuitive.

TABLEAU 1.3 – processus de fonctionnement de l'IOT [24]

1.3.4 Architecture d'un système d'IoT

L'architecture IoT comprend plusieurs briques de systèmes IoT connectés pour garantir que les données des objets générées par les capteurs sont collectées, stockées et traitées dans les *entrepôts de données* et que les actionneurs des objets exécutent les commandes envoyées via une application utilisateur. [25]

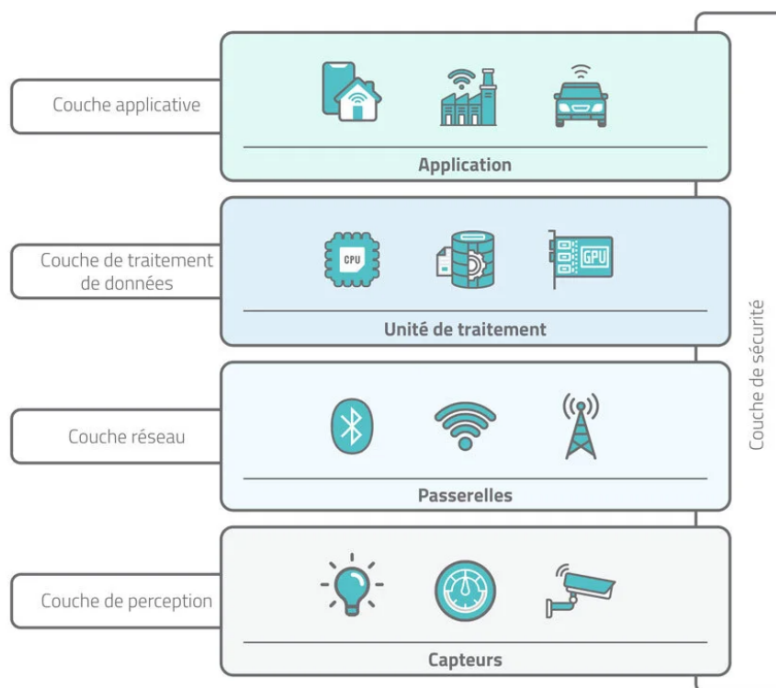


FIGURE 1.4 – Architecture de l'IOT

[25]

La Figure 1.4 présente de manière visuelle les différentes couches de l'Internet des objets (IoT), qui sont définis comme suit :

1. **Couche perception** : La couche de perception est responsable de la conversion des signaux analogiques en données numériques et vice versa. Il s'agit de l'étape initiale de tout système IoT, englobant un large éventail d'objets qui servent de pont entre les mondes réel et numérique. Ces objets IoT peuvent être classés en plusieurs catégories : capteurs, actionneurs, machines et dispositifs. [25]
2. **Couche réseau** : Les données collectées par ces appareils doivent être transmises et traitées. C'est le rôle de la couche réseau, qui assure la connexion entre ces appareils, d'autres objets intelligents, les serveurs et les équipements réseau. Elle prend également en charge la transmission de toutes les données. Les communications entre les appareils, les services cloud et les passerelles reposent sur différentes technologies : Ethernet, réseaux cellulaires, LPWAN, WiFi. [25]
3. **Couche de traitement des données** : Cette couche stocke et traite les données issues de la couche réseau via des plateformes IoT. Son fonctionnement repose sur deux étapes principales : - *Accumulation des données* : Capture des données en temps réel via une API et stockage pour une utilisation ultérieure. - *Abstraction des données* : Préparation et transformation des données pour qu'elles puissent être exploitées par les applications. [25]
4. **Couche applicative** : Cette couche constitue l'interface utilisateur finale, où les données traitées sont présentées sous une forme exploitable. Elle permet l'interaction entre les utilisateurs et le système IoT à travers des applications spécifiques selon le domaine (santé, industrie, agriculture, etc.).[25]

1.3.5 Avantages et inconvénients de l'IoT :

1.3.5.1 Avantages :

-**Analyse en temps réel** : Permet des décisions rapides et éclairées en exploitant les données des appareils connectés. [26]

-**Productivité accrue** : Automatisation des tâches pour une meilleure efficacité, tant personnelle que professionnelle. [26]

-**Commodité au quotidien** : Facilite les routines (gestion des thermostats, rappels, maintenance automatisée). [26]

-**Amélioration de la prise de décision** : Analyse avancée pour optimiser les processus et réduire les coûts. [26]

1.3.5.2 Inconvénients

-**la sécurité** : l'un des plus gros points faibles de l'IOT est la sécurité, en raison de l'énorme quantité de données manipulée, ce qui peut engendrer des pertes financières considérables et qui

nuire à la réputation des entreprises .[24]

- **la confidentialité** : en cas de cyberattaque , le probabilité que les informations concernant l'entreprise (clients , employés..ect) soient modifiées , supprimé ou volées est extrêmement élevée , et ceci concerne davantage les systèmes IOT car toutes les données sont partagées .[24]

-**Les défaillances** : un système IOT qui gère des biens matériels comme une voiture intelligente ou une pompe à insuline , pourrait provoquer des conséquences désastreuses s'il venait à se produire une panne ou une défaillance quelconque [24].

1.3.6 Capteurs de surveillance météorologique

Ce tableau présente une vue d'ensemble des différents types de capteurs utilisés pour la surveillance météorologique :

Capteur	Description	Image
DHT22	Capteur de mesure d'humidité et de température de l'air à faible coût. Il comprend une thermistance pour la température et un capteur d'humidité capacitif [27].	
GS3 Sensor	Mesure la teneur en eau, la température et la conductivité électrique (EC) à 70 MHz pour minimiser les effets de salinité et de texture. Sonde robuste à aiguilles en acier inoxydable [28].	
DHT11	Capteur numérique de température et d'humidité de base, très économique. Utilise un capteur capacitif et une thermistance, avec sortie numérique simple [29].	
Anémomètre	Le système Alarnemo combine anémomètre et émetteur avec affichage, sorties analogiques et numériques pour vitesse/direction du vent et alarmes [30].	

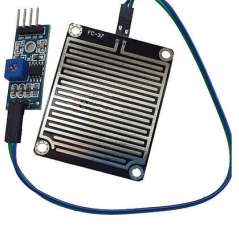
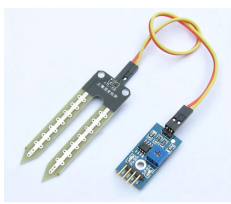
Capteur	Description	Image
YL-83	Détecte les gouttes d'eau et l'humidité. La sortie analogique est connectable à un microcontrôleur pour estimer l'intensité des précipitations [31].	
Apogee SP-110	Capteur analogique auto-alimenté (0 à 400 mV), avec photodiode en silicium dans un boîtier autonettoyant et robuste [32].	
YL-69	Capteur d'humidité du sol pour systèmes d'irrigation ou de surveillance. Composé d'une carte électronique et d'une sonde à deux plots [33].	

TABLEAU 1.4 – Capteurs utilisés pour la surveillance environnementale et agricole

1.4 Comparaison des travaux existants

1.4.1 prévision météorologique

1.4.1.1 Travail de : Yashon O. Ouma et al.

Cette étude compare le réseau neuronal LSTM et le réseau neuronal à ondelettes (WNN) pour la prévision spatio-temporelle des tendances des séries chronologiques de précipitations et de ruissellement dans des bassins hydrologiques peu jaugés. À partir de données d'observation in situ à long terme sur 30 ans (1980-2009)[34].

1.4.1.2 Travail de : Nitin Singh et al.

L'objectif de ce travail est de développer une application de prévision météorologique peu coûteuse, en utilisant le concept d'apprentissage automatique en Python sur la carte Raspberry Pi, pour améliorer le domaine de l'agriculture dans des zones reculées en exploitant le potentiel de l'Internet des objets (IOT) pour la surveillance météorologique en temps réel et à distance en utilisant des capteurs connectés [35].

1.4.1.3 Travail de : Pan, B., Hsu, K., AghaKouchak, A., & Sorooshian, S.

Cette étude présente un modèle de CNN permettant de lisser cet aspect de l'écart type des prévisions pluviométriques quotidiennes. Plus précisément, ce modèle est limité aux prévisionnistes dont les variables sont résolues directement par discrétisation des équations de la dynamique atmosphérique. Il propose également des méthodes simples de visualisation et d'analyse pour interpréter les modèles et leurs résultats. Cette étude apporte deux contributions : premièrement, elle propose une nouvelle approche pour améliorer l'estimation numérique des précipitations ; deuxièmement, le modèle proposé offre des implications importantes pour l'amélioration des schémas de paramétrisation des précipitations grâce à une approche basée sur les données [36].

Travail (Auteur/Année)	Dataset	Technique	Critères de performance	Points forts	Limitations
Yashon O.Oumaet et al.,(2021)[34]	-données d'observation in situ 1980-2009 NCEP-CFSR	- LSTM	R2 = 0.86	-LSTM-RNN capable de maintenir l'état et de propager les gradients de manière stable sur de longues périodes. -Le LSTM est plus performant que le WNN en terme de précision.	-La prédiction s'effectue à l'échelle mensuel pas journalière.
Nitin Singh et al., (2019) [35].	Delhi Weather Data	Random Forest Classification	-Accuracy (Précision) = 87,90	-Interface graphique utilisateur pour simplifier l'interaction. -Systeme de prévision météorologique peu coûteuse -Utilise l'iot pour connecter les capteurs de météorologique	Aucune mise en relation avec d'autres méthodes de deep learning ou modèles météorologiques sophistiqués. -Dépendance vis-à-vis des données climatiques passées d'une région spécifique (Delhi)

Baoxiang Pan et al., (2019) [36]	Centre de prévision climatique de la National Oceanic and Atmospheric Administration (CPC) NARR (North American Regional Reanalysis) période : 1979 à 2017	-CNN	$r = 0.84$ RMSE = 4.83	-robuste face aux différentes initialisations de paramètres -CNN évite les deux faiblesses majeures des méthodes downsampling statistics classique -Hiérarchique, encode les motifs spatiaux dynamiques	-Problème d'interprétabilité : Les modèles de DL sont souvent considérés comme "boîtes noires". -limité au USA non testé ailleurs -performance limitée dans le centre des états - unis -Dépendance à la qualité et la diversité des données d'entraînement.
----------------------------------	--	------	---------------------------	---	--

TABLEAU 1.5 – Résumé du travail sur la prévision météorologique utilisant l'apprentissage automatique

1.4.2 prévision d'irrigation

1.4.2.1 Travail de : Youness Tace, Mohamed Tabaa, Sanaa Elfilali, Cherkaoui Leghris, Hassna Bensag, Eric Renault

Cet article présente une solution d'irrigation intelligente et économique, adaptée à divers contextes agricoles. En s'appuyant sur des algorithmes de machine learning, cette approche utilise des capteurs (humidité du sol, température, pluie) pour collecter des données sur plusieurs mois, avec la plateforme Node-RED et la base de données MongoDB. Plusieurs modèles ont été testés. Une application web a été développée pour visualiser les données et les prédictions, facilitant la supervision et la prise de décision[37].

Travail (Auteur/Année)	Dataset	Technique	Critères de performance	Points forts	Limitations
Youness Tace et al.(2022) [37]	Données collectées (Node RED, MongoDB)	KNN	$R^2 = 0.983$ $RMSE = 0.12$	Approche flexible et faible coût, adaptable à différents contextes, reconnaissance élevée du KNN	Les autres modèles ont des taux de reconnaissance plus faibles, dépendance à la qualité des données collectée

TABLEAU 1.6 – Résumé du travail sur la prévision d’irrigation utilisant l’apprentissage automatique

1.5 conclusion

Dans ce chapitre, nous avons commencé par exposer les concepts fondamentaux relatifs à l’Internet des Objets et à l’agriculture de précision. Par la suite, en examinant les travaux antérieurement effectués, nous avons réussi à saisir plus clairement les méthodes utilisées et leurs restrictions courantes. Ces observations nous ont aidés à déterminer des voies d’amélioration tangibles pour notre projet. Cette phase initiale nous prépare donc à aborder, dans le prochain chapitre, la conception de notre système.

Chapitre 2

Conception du système

2.1 Introduction

Ce chapitre est dédié à la conception d'un système d'irrigation intelligent, basé sur l'Internet des Objets et l'intelligence artificielle. Il représente les étapes clés, allant de l'estimation des précipitations via des algorithmes d'apprentissage automatique, à l'élaboration d'un système de prise de décision intégrant des éléments agronomiques et environnementaux pour optimiser l'irrigation. L'objectif est de proposer une solution qui adapte automatiquement les techniques d'irrigation selon les exigences véritables des plantations, tout en réduisant au minimum la consommation d'eau.

2.2 Motivation de l'étude de cas

L'agriculture constitue la principale source de la production alimentaire dans notre pays. Elle contribue à plus de 12% du produit intérieur brut (PIB). Le développement de ce secteur devient de plus en plus indispensable pour assurer une autosuffisance alimentaire . Cependant, l'absence de contrôle de l'irrigation a montré qu'elle peut entraîner un épuisement des ressources et conduit parfois à la détérioration des récoltes à cause de la sur-irrigation ou sous-irrigation. [38]

Le système d'irrigation intelligent permet d'économiser l'eau en arrosant uniquement lorsque cela est nécessaire, tout en assurant une meilleure santé des cultures grâce à un apport adapté. Grâce aux capteurs et aux données météorologiques, les agriculteurs peuvent prendre des décisions plus facilement et avec précision, en leur fournissant des informations claires sur l'état de leur champ.

2.3 Processus de conception du système

Afin d'assurer une gestion optimale de l'irrigation et de favoriser la croissance des plantes, nous exploitons les opportunités offertes par l'Internet des objets (IoT) ainsi que les algorithmes d'apprentissage automatique. Ce système collecte en temps réel des données météorologiques et agronomiques à l'aide de capteurs IoT. Ces données sont ensuite traitées par des algorithmes d'apprentissage automatique afin de prédire les précipitations. Les résultats de ces prédictions, combinés à d'autres variables collectées, servent d'entrées pour un système d'aide à la décision, dont la sortie permet de déterminer les besoins en irrigation de manière précise et efficace. La FIGURE 2.1 illustre visuellement le processus de conception de notre système.

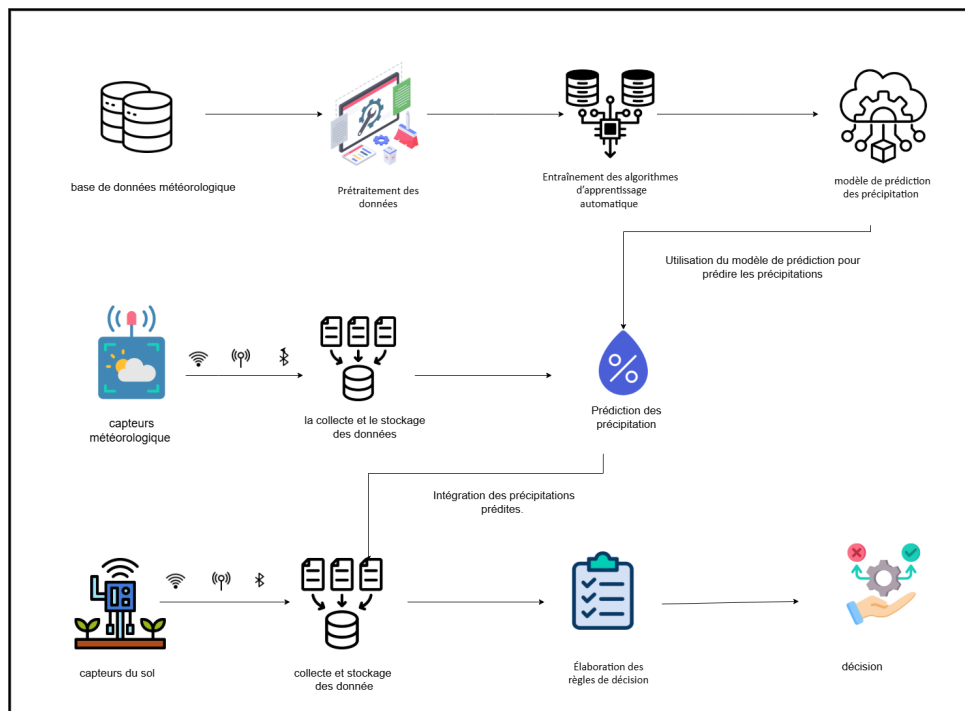


FIGURE 2.1 – Processus de conception du système

La première étape consiste à collecter les données météorologiques à partir d'un jeu de données à cet effet. Ce jeu de données a été choisi en fonction de la qualité de ses données et de sa pertinence. Ces dernières ont été enrichies en incluant les données de la plateforme ouverte open-meteo.com [39], et nous avons rajouté des colonnes qui nous intéressent, telles que l'humidité relative, la pression atmosphérique et le point de rosée, qui sont utilisées pour les besoins de l'agriculture. Ensuite, nous avons préparé les données en veillant à la normalisation des entrées, à la détection des valeurs aberrantes et au contrôle de leur qualité, afin d'assurer une base solide pour l'analyse.

Plusieurs algorithmes d'apprentissage automatique ont été entraînés pour prédire les précipitations, et le modèle le plus performant a été retenu. Les données d'entrée nécessaires au fonctionnement du système de décision en irrigation ont été acquises et structurées. Les prévi-

sions de précipitations issues du modèle ont été intégrées comme variable d'entrée du système d'irrigation. Des règles de décision ont été définies et les variables de sortie du système ont été spécifiées selon les besoins agronomiques. Enfin, le système a été validé et évalué afin de vérifier sa performance et sa fiabilité.

2.4 Prévision météorologique

2.4.1 Dataset

Algiers Weather Data Cet ensemble de données fournit des données météorologiques historiques pour Alger, la capitale de l'Algérie, couvrant la période de janvier 2002 à août 2023. Ces données incluent diverses variables météorologiques pour aider les chercheurs, les analystes et les passionnés de météorologie à explorer et analyser les tendances et les conditions climatiques à Alger au fil des ans. [40]

Le choix du jeu de données météorologiques utilisé dans notre étude a été guidé par plusieurs considérations essentielles. Nous avons privilégié un dataset météorologique pour sa pertinence dans le contexte agricole, contenant des variables clés telles que les précipitations, la température et la vitesse du vent.

Afin d'enrichir ce jeu de données initial, **une fusion** avec des données issues de la plateforme open source **Open Meteo.com** a été réalisée. Cette fusion a permis d'ajouter des colonnes supplémentaires telles que **l'humidité relative, la pression atmosphérique et le point de rosée**, renforçant ainsi la qualité et la profondeur des informations disponibles pour l'analyse prédictive.

Le volume du dataset météorologique enrichi atteint **7913 lignes** pour une taille de **1,22 Mo**, couvrant différentes périodes et conditions climatiques.

Le tableau ci-dessous présente une description détaillée de ce jeu de données.

Attribut	Description
date	Date de l'observation (format ISO 8601).
weathercode (WMO code)	Code météorologique selon la norme WMO décrivant les conditions (ex. ciel clair, pluie, brouillard, etc.).
temperature_2m_max (°C)	Température maximale journalière à 2 mètres du sol (en degrés Celsius).
temperature_2m_min (°C)	Température minimale journalière à 2 mètres du sol (en degrés Celsius).
temperature_2m_mean (°C)	Température moyenne journalière à 2 mètres du sol (en degrés Celsius).
apparent_temperature_max (°C)	Température maximale ressentie (en degrés Celsius).

Tableau 2.1 – suite

Attribut	Description
apparent_temperature_min (°C)	Température minimale ressentie (en degrés Celsius).
apparent_temperature_mean (°C)	Température moyenne ressentie (en degrés Celsius).
sunrise	Heure du lever du soleil (format ISO 8601).
sunset	Heure du coucher du soleil (format ISO 8601).
precipitation_sum (mm)	Somme des précipitations quotidiennes (en millimètres).
rain_sum (mm)	Quantité de pluie tombée (en millimètres).
snowfall_sum (cm)	Quantité de neige tombée (en centimètres).
precipitation_hours (h)	Nombre d'heures de précipitation dans la journée.
windspeed_10m_max (km/h)	Vitesse maximale du vent mesurée à 10 mètres (en kilomètres par heure).
windgusts_10m_max (km/h)	Vitesse maximale des rafales à 10 mètres (en kilomètres par heure).
winddirection_10m_dominant (°)	Direction dominante du vent à 10 mètres (en degrés).
shortwave_radiation_sum (MJ/m ²)	Rayonnement solaire cumulé à la surface (en mégajoules par mètre carré).
et0_fao_evapotranspiration (mm)	Évapotranspiration de référence selon la méthode FAO (en millimètres).
relative_humidity_2m (%)	Humidité relative moyenne à 2 mètres du sol (en pourcentage).
dew_point_2m (°C)	Température du point de rosée à 2 mètres (en degrés Celsius).
pressure_msl (hPa)	Pression atmosphérique au niveau de la mer (en hectopascals).

TABLEAU 2.1 – Description des attributs du jeu de données
météorologiques

2.4.2 Prétraitement des Données

Dans cette section, nous avons concentré notre attention sur le prétraitement des données, une phase indispensable pour garantir la crédibilité et l'organisation adéquate des informations employées. Cette étape a pour but de rectifier les incohérences, de standardiser les formats et de

préparer les données en vue d'analyses à venir. Nous exposons ci-dessous les diverses phases mises en application durant cette procédure.

2.4.2.1 Enrichissement des données

Pour assurer la qualité des prédictions météorologiques, nous avons constaté que certaines variables indispensables n'étaient pas présentes dans notre jeu de données. Afin de compléter ces informations manquantes, nous avons intégré des données issues du site Open-Meteo.com, source également utilisée pour le jeu de données principal. Cette homogénéité a permis une fusion cohérente des deux ensembles, Le processus de fusion s'est déroulé selon les étapes suivantes :

- **Préparation du jeu de données secondaire** Nous avons d'abord nettoyé les données issues du second fichier en convertissant les colonnes vers des types appropriés. La colonne temporelle a été transformée au format `datetime`, et les autres variables numériques ont été corrigées si nécessaire.

- **Agrégation à l'échelle journalière** Les observations initialement horaires ont été regroupées par date, en calculant la moyenne journalière pour chaque variable afin de lisser les fluctuations et structurer les données autour d'une échelle temporelle quotidienne.

- **Chargement et harmonisation du fichier principal** : Le jeu de données principal a été ensuite importé. La colonne de date a été renommée et convertie au bon format afin d'assurer une correspondance avec le second dataset.

- **Fusion des jeux de données** : Une jointure interne a été effectuée sur la colonne de date. Cette fusion de type interne a permis de ne conserver que les dates communes aux deux jeux, garantissant ainsi une cohérence temporelle dans les enregistrements.

- **Export des données enrichies** : Le fichier fusionné a été enregistré pour servir de base à la suite de nos analyses et expérimentations.

2.4.2.2 Normalisation des données

Nous avons procédé à une normalisation des variables numériques afin d'uniformiser leur échelle et d'éviter qu'une variable à forte amplitude n'influence excessivement les modèles d'apprentissage. Pour cela, nous avons utilisé la méthode de standardisation via l'outil *StandardScaler* de la bibliothèque *scikit-learn*. Cette approche permet de centrer chaque variable autour de zéro et de la ramener à une variance unitaire. Nous avons aussi utilisé la technique *MinMaxScaler*, particulièrement pour normaliser les variables dans une plage de valeurs allant de 0 à 1, lorsque cela s'avérait plus pertinent pour la stabilité de l'entraînement du modèle.

2.4.2.3 Nettoyage des données

Nous avons procédé à une vérification de l'intégrité des données après leur enrichissement. L'analyse a montré qu'aucune valeur manquante n'était présente dans les champs observés, ce

qui nous a permis d'écarter toute étape supplémentaire de traitement, telle que l'imputation.

Sur les 22 caractéristiques de départ, les variables « sunrise », « sunset » et « precipitation-hours (h) » ont été éliminées en raison de leur absence de pertinence pour la prédiction. De plus, les colonnes "rain-sum (mm)" et "snowfall-sum (cm)" n'ont pas été considérées, car elles sont déjà stockées dans la variable cible "precipitations-sum (mm)", qui représente la somme totale des précipitations (rain + snow).

pour traiter les valeurs aberrantes nous avons utilisé la méthode de l'intervalle interquartile (IQR) , Cette méthode consiste à identifier les valeurs situées en dehors de l'intervalle défini par $Q_1 - 1.5 \times IQR$ et $Q_3 + 1.5 \times IQR$, où Q_1 et Q_3 représentent respectivement les premier et troisième quartiles. Ensuite , les valeurs aberrantes ont été remplacées par des valeurs manquantes , puis interpolées selon l'axe temporel afin de préserver la continuité des séries . Enfin, les données ont été ajustées (clipping) à de nouvelles bornes pour garantir la cohérence statistique . Ce traitement nous a permis d'obtenir un jeu de données plus propre et plus fiable.

2.4.3 Construction du Modèle de Prédiction Météorologique

La phase essentielle de l'élaboration du modèle prédictif représente le cœur de notre recherche, qui a pour but d'implémenter un système d'irrigation capable d'évaluer les besoins en eau et d'assurer une santé optimale des cultures. Cette étape, qui est cruciale pour notre approche, s'appuie sur la sélection adéquate d'algorithmes d'apprentissage automatique. Quatre méthodes algorithmiques ont été choisies, toutes possédant des traits distinctifs et contribuant de manière spécifique à la réalisation de notre but général :

2.4.3.1 Random forest

Le Random Forest (qui signifie *forêt aléatoire*) est un ensemble d'arbres de décision utilisés pour prédire une quantité ou une probabilité. un Random Forest a besoin de trois hyper-paramètres principaux (paramètres fixes), qui doivent être définis avant l'entraînement. Il s'agit notamment de la taille des arbres (le nombre de nœuds maximal), du nombre d'arbres à utiliser et le nombre de caractéristiques échantillonnées (nombre de variables aléatoires choisies à chaque mélange depuis les variables explicatives). À partir de là, le modèle peut être utilisé pour résoudre les problèmes de régression ou de classification. [41]

Dans le cadre de notre recherche, nous avons élaboré un modèle de régression basé sur une forêt aléatoire comprenant 100 arbres décisionnels, dans le but de prévoir les précipitations. Le calcul de la précipitation prédite $\hat{y}$ pour un échantillon x s'effectue de la manière suivante :

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T h_t(x) \quad (2.1)$$

où :

— $\hat{y}$: la précipitation prédite (en millimètres) pour un échantillon donné.

- T : le nombre total d'arbres de décision dans la forêt .
- t : l'indice de chaque arbre de décision, avec $t = 1, 2, \dots, T$.
- $h_t(x)$: la prédiction de l'arbre t pour l'échantillon x .
- x : le vecteur des variables météorologiques d'entrée (température, humidité, pression, vent, etc.) utilisées pour estimer la précipitation.

2.4.3.2 Support Vector Regressor (SVR)

La régression à vecteurs de support (SVR) est une technique d'apprentissage automatique utilisée pour les tâches de régression. Variante des machines à vecteurs de support (SVM), elle est conçue pour prédire des valeurs numériques continues . SVR vise à trouver une fonction qui prédit une variable cible continue tout en maximisant la marge entre les valeurs prédites et les points de données réels. Elle identifie une “marge” autour de la ligne de régression et s'ajuste en minimisant l'erreur. Les “vecteurs de support” déterminent le modèle. L'astuce du noyau permet de modéliser des relations non linéaires, avec réglage des hyperparamètres (C, noyau). La fonction de perte insensible à epsilon tolère certaines erreurs et pénalise celles hors marge. [42]

Dans notre recherche, dont l'objectif est de prédire les précipitations, , nous avons utilisée un noyau (kernel='rbf')(Radial Basis Function) pour capter les relations non linéaires , La formule qui définit ce noyau est la suivante :

$$K(x_i, x) = \exp(-\gamma \|x_i - x\|^2) \quad (2.2)$$

où :

- x : vecteur des caractéristiques de l'échantillon à prédire,
- x_i : vecteur des caractéristiques d'un échantillon d'entraînement,
- $\|x_i - x\|^2$: distance euclidienne au carré entre x_i et x ,
- γ : paramètre du noyau contrôlant l'influence d'un seul point d'entraînement,
- $\exp$: fonction exponentielle.

de plus , un Paramètre de régularisation (C=1) et un Tolérance d'erreur insensible (epsilon=0.1)

Le modèle SVR prédit la précipitation $\hat{y}$ à partir d'un ensemble de caractéristiques météorologiques en utilisant une fonction noyau K . La formule est donnée par :

$$\hat{y}(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \cdot K(x_i, x) + b \quad (2.3)$$

où :

- $\hat{y}(x)$: valeur prédite de la variable cible (précipitation) pour un échantillon x ,
- n : nombre d'échantillons d'entraînement,

- α_i, α_i^* : coefficients appris par le modèle SVR pour chaque vecteur de support,
- $K(x_i, x)$: fonction noyau mesurant la similarité entre x_i (échantillon d'entraînement) et x (échantillon à prédire),
- b : terme de biais ajusté lors de l'entraînement.

2.4.3.3 LSTM simple(Long Short-Term Memory)

est un type amélioré de réseau neuronal récurrent. L'objectif principal de leur conception était de résoudre les problèmes persistants des réseaux neuronaux récurrents et d'améliorer les résultats obtenus. Les réseaux LSTM ont été spécialement conçus pour surmonter les difficultés telles que le problème du gradient qui disparaît ou explose lors de l'apprentissage des réseaux neuronaux RNN classiques .[43]

Grâce à leur architecture unique, les LSTM sont capables de capturer et de gérer les dépendances à long terme dans les séquences en utilisant des portes de contrôle qui régulent le flux d'informations et maintiennent une mémoire à long terme .[43]

Dans le cadre de ce travail , un modèle de prévision des précipitations a été créé en s'appuyant sur une architecture **de réseau de neurones à mémoire longue courte durée (LSTM)**. Ce modèle a pour objectif de produire **des prévisions sur une durée de 7 jours**, à partir d'une séquence d'observations météorologiques passées. L'architecture mise en place comprend **une première couche LSTM comprenant 128 neurones**, dont le rôle est de capturer les dépendances temporelles présentes dans les données d'entrée. Dans le but de minimiser le risque de surapprentissage, **une couche de régularisation de type Dropout**, avec un taux de 30 % , est appliquée. La sortie est par la suite orientée vers **une deuxième couche LSTM, formée de 64 neurones** , qui condense les informations extraites. Enfin, **une couche Dense composée de 7 neurones** est chargée de générer directement les valeurs attendues de précipitations pour chaque jour de la semaine future.

L'entraînement du modèle a été réalisé à partir de séries temporelles historiques, puis validé sur des données non vues, permettant de prédire avec efficacité les précipitations journalières pour une semaine.

2.4.3.4 LSTM Profond

Les architectures de réseaux neuronaux récurrents (RNN), comme les réseaux LSTM, sont conçues pour résoudre le problème de gradient nul qui survient fréquemment lors de l'entraînement des RNN conventionnels sur de longues séquences de données. En empilant les unités LSTM les unes sur les autres, les réseaux LSTM profonds s'appuient sur la conception fondamentale des LSTM pour capturer des schémas et des relations plus complexes dans les données séquentielles. [44]

Un réseau LSTM profond est construit à partir de plusieurs couches LSTM superposées. Chaque couche LSTM comprend trois types de portes : d'entrée, d'oubli et de sortie, ainsi qu'une

séquence de cellules mémoire qui conservent les informations au fil du temps. En contrôlant les informations entrantes et sortantes des cellules mémoire, ces portes permettent aux réseaux LSTM de mettre à jour et d'utiliser les données stockées de manière sélective, selon les besoins. [44]

L'architecture du réseau **LSTM profond** est illustrée dans la figure **FIGURE 2.2**

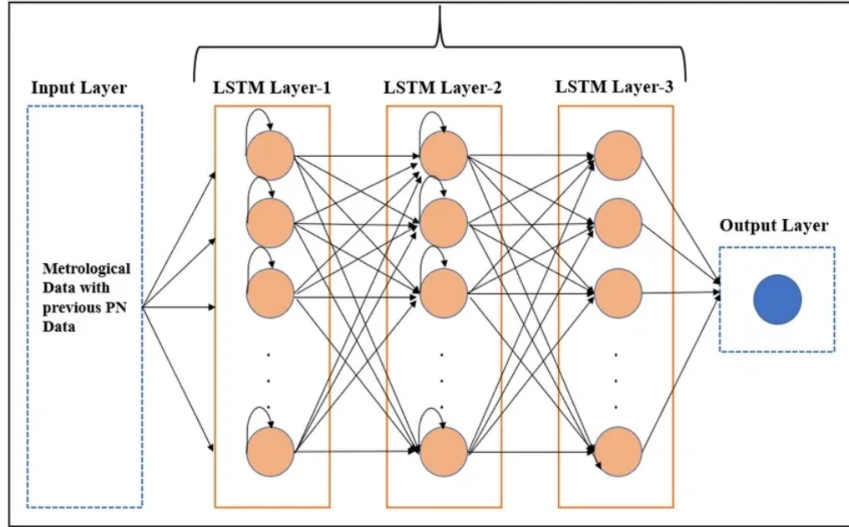


FIGURE 2.2 – Architecture de LSTM profond

[45]

Dans ce travail, nous avons utilisé une architecture LSTM profonde, composée de plusieurs couches empilées afin de capturer des dépendances temporelles complexes sur différentes échelles. Plus précisément, l'architecture contient quatre couches LSTM avec 256, 128, 64 et 32 unités respectivement. Les trois premières couches retournent l'ensemble de la séquence, ce qui permet aux couches supérieures de continuer à apprendre sur toute la série temporelle. La dernière couche LSTM, plus compacte, ne conserve que l'état final, servant ainsi de représentation résumée de la séquence. Cette sortie est transmise à une couche dense comprenant 7 neurones, adaptée à la prédiction de plusieurs valeurs cibles (comme une série de 7 jours à venir).

Un réseau LSTM (*Long Short-Term Memory*) traite une séquence d'entrée x_t à chaque pas de temps t , en mettant à jour un état caché h_t et un état mémoire C_t , à l'aide de différentes portes.

Les équations internes d'une cellule LSTM sont les suivantes :

— **Porte d'oubli :**

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.4)$$

— **Porte d'entrée :**

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.5)$$

— **État candidat :**

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.6)$$

— **Mise à jour de la mémoire :**

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (2.7)$$

— **Porte de sortie :**

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.8)$$

— **État caché (sortie) :**

$$h_t = o_t \odot \tanh(C_t) \quad (2.9)$$

où :

- σ : fonction sigmoïde
- $\tanh$: tangente hyperbolique
- $\odot$: produit élément par élément (Hadamard)
- W_*, b_* : matrices de poids et vecteurs de biais
- h_{t-1} : état caché au temps précédent
- C_{t-1} : état mémoire au temps précédent

Pour réduire les risques de surapprentissage, plusieurs couches Dropout ont été insérées à différents niveaux du réseau, avec des taux établis à 30 %, 0 % et 20 % selon les étapes. Cet algorithme est reconnu pour ajuster automatiquement les coefficients d'apprentissage à chaque itération, en fonction de l'historique des gradients. Pour la fonction de perte, on a choisi celle de Huber en raison de sa performance à traiter les données avec des valeurs extrêmes.

La fonction de Huber combine les avantages de l'erreur quadratique moyenne (MSE) et de l'erreur absolue moyenne (MAE). Elle est définie comme suit :

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{si } |y - \hat{y}| \leq \delta \\ \delta \cdot (|y - \hat{y}| - \frac{1}{2}\delta) & \text{sinon} \end{cases} \quad (2.10)$$

où :

- y : vraie valeur (valeur cible)
- $\hat{y}$: prédiction du modèle
- δ : seuil à partir duquel la fonction passe de quadratique à linéaire (souvent $\delta = 1$)

2.4.4 Evaluation des Modèles et Sélection du Meilleur

L'évaluation de l'efficacité des modèles prédictifs en matière de précipitations se base sur des métriques appropriées. Quatre modèles ont été mis en parallèle dans notre recherche, en utilisant des critères tels que le coefficient de détermination **R²**, l'Erreur Quadratique Moyenne (**MSE**) et l'Erreur Absolue Moyenne (**MAE**). Le **LSTM Profond** s'est montré le meilleur, alliant précision et stabilité, avec un **R² de 96%**, une **MAE de 1.39%** et une **MSE de 0.07 %**, ce qui en fait une solution particulièrement adaptée aux besoins de la prévision météorologique. Ces résultats sont présentés et analysés en détail dans le chapitre 3.

2.5 Système d'Irrigation

2.5.1 Paramètres d'entrée du système

Le système d'irrigation intelligent repose sur plusieurs paramètres d'entrée, essentiels pour prendre une décision adaptée et contextuelle concernant les besoins en eau des cultures. Ces paramètres proviennent à la fois de sources de prédiction météorologique, d'autres sont captées en temps réel par des capteurs IoT déployés dans les champs

1. **Précipitation** : Elles jouent un rôle déterminant dans la planification de l'irrigation, en évitant les arrosages inutiles en cas de pluie imminente.
2. **Humidité du sol** : Paramètre direct de décision. Un seuil critique d'humidité permet de déclencher l'irrigation si nécessaire.
3. **pH du sol** : Bien que le pH n'affecte pas directement l'irrigation, il est intégré pour orienter des recommandations plus précises sur les cultures à venir ou les besoins correctifs.
4. **type de sol** : Le système distingue notamment les sols légers et lourds, chacun ayant une capacité de rétention d'eau différente, ce qui influence la fréquence et la quantité d'irrigation.
5. **stade de développement** : Les besoins en eau d'une plante varient selon son cycle de croissance.
6. **besoin journalier en eau** : représente le volume d'eau que la plante doit recevoir quotidiennement (par arrosage ou précipitations) pour garantir un développement optimal sans être exposée à un stress hydrique.

Consommation journalière en eau du blé		
Montaison	Épiaison - Floraison	Formation du grain
3,5 à 4 mm/jour	6 mm/jour	7,5 à 8 mm/jour

TABLEAU 2.2 – Consommation d'eau journalière moyenne selon les stades du blé[46]

7. **Jours après semis** : Le Jour Après Semis est une référence temporelle qui dénote le nombre de jours passés depuis la date de semis d'une culture. voici ce tableau qui définit chaque stade a son intervalle en jours après semis

Stade phénologique	Intervalle en JAS (approximatif)
Germination	0–10 jours
Levée	10–25 jours
Tallage	25–60 jours
Montaison	60–110 jours
Épiaison	110–125 jours
Floraison	125–135 jours
Grain laiteux	135–150 jours
Grain pâteux	150–170 jours
Maturité	170–180 jours

TABLEAU 2.3 – Correspondance entre stade phénologique du blé et jours après semis[47]

La figure suivante illustre les stades de croissance du blé.

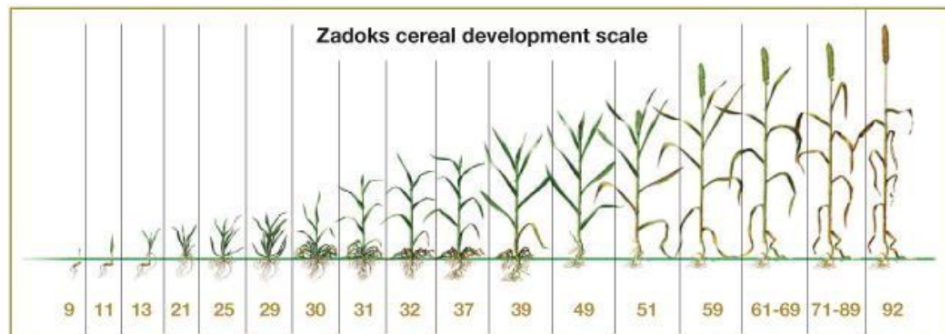


FIGURE 2.3 – les stades de croissance du blé.

[45]

2.5.2 Algorithme de déclenchement de l'irrigation

L'algorithme de déclenchement est la partie fonctionnelle du système qui met en œuvre les règles de décision à partir des paramètres collectés. Il permet de déterminer quand irriguer, combien d'eau utiliser et pendant combien de temps.

Algorithm 1 Système de décision basé sur les conditions agro-climatiques

```
1: Entrées :
2:  $E \leftarrow$  Stade Zadoks
3:  $J \leftarrow$  Jours après semis
4:  $H \leftarrow$  Humidité du sol (%)
5:  $pH \leftarrow$  pH du sol
6:  $T \leftarrow$  Température maximale journalière (°C)
7:  $S \leftarrow$  Type de sol (léger ou lourd)
8:  $R \leftarrow$  Précipitations prévues (mm)
9:  $B \leftarrow$  Besoin journalière en eau (mm)

10: Seuils définis :
11: if  $S = \text{léger}$  then
12:    $H_{seuil} \leftarrow 40$ 
13: else
14:    $H_{seuil} \leftarrow 50$ 
15: end if
16:  $T_{critique} \leftarrow 25$ 
17: Intervalle optimal du pH :  $5.5 \leq pH \leq 5.8$ 
18: Période critique pour floraison :  $125 \leq J \leq 132$ 

19: Début des règles de décision :
20: if  $pH < 5.5$  or  $pH > 5.8$  then
21:   Alerte : pH hors plage optimale (5.5–5.8)
22: end if
23: if ( $E = \emptyset$ ) or ( $J = \emptyset$ ) or ( $H = \emptyset$ ) or ( $pH = \emptyset$ ) or ( $T = \emptyset$ ) or ( $S = \emptyset$ ) or ( $R = \emptyset$ )
   then
24:   Alerte : Donnée capteur manquante ou invalide détectée — Vérification requise
25: end if

26: Stades : GERMINATION, LEVÉE, TALLAGE (JAS 0–60)
27: if  $E = \text{Germination}$  or  $E = \text{Levée}$  or  $E = \text{Tallage}$  then
28:   if  $R < B$  and  $H < H_{seuil}$  then
29:     Décision : Irrigation
30:     Justification : Pluie insuffisante et sol sec
31:   else if  $R \geq B$  or  $H \geq H_{seuil}$  then
32:     Décision : Pas d'irrigation
33:     Justification : Apports naturels suffisants
34:   end if
35:   if  $H \geq H_{seuil}$  and  $T > T_{critique}$  then
36:     if  $S = \text{léger}$  then
37:       Décision : Pas d'irrigation
38:       Justification : Sol humide mais risque d'assèchement rapide
       —Surveillance renforcée
39:     else
40:       Décision : Pas d'irrigation
41:       Justification : Température élevée mais sol lourd retient bien l'eau
       Surveillance simple
42:     end if
43:   end if
44: end if
```

Suite d’algorithm 1

```
1: Stades : MONTAISON ou ÉPIAISON (JAS 60–125)
2: if  $E = \text{Montaison}$  or  $E = \text{Épiaison}$  then
3:   if  $R < B$  and  $H < H_{seuil}$  then
4:     Décision : Irrigation
5:     Justification : Déficit hydrique en stade critique
6:   else if  $R \geq B$  and  $H \geq H_{seuil}$  then
7:     Décision : Pas d’irrigation
8:     Justification : Pluie suffisante + sol humide
9:   end if
10:  if  $H \geq H_{seuil}$  and  $T > T_{critique}$  then
11:    if  $S = \text{léger}$  then
12:      Décision : Irrigation
13:      Justification : Température critique + sol sableux = risque d’assèchement
14:    else
15:      Décision : Pas d’irrigation
16:      Justification : Sol lourd + humidité suffisante — Surveillance recommandée
17:    end if
18:  end if
19: end if

20: Stade : FLORAISON (JAS 125–135)
21: if  $E = \text{Floraison}$  and  $125 \leq J \leq 132$  then
22:   Décision : Pas d’irrigation
23:   Justification : Interdiction d’irriguer durant la sortie des étamines
24: else if  $E = \text{Floraison}$  and  $J > 132$  then
25:   if  $R < B$  and  $H < H_{seuil}$  then
26:     Décision : Irrigation
27:     Justification : Déficit hydrique post-étamines
28:   else if  $H \geq H_{seuil}$  and  $T > T_{critique}$  then
29:     if  $S = \text{léger}$  then
30:       Décision : Irrigation
31:       Justification : Sol sableux + température élevée → évaporation rapide → risque
32:       de stress post-floraison
33:     else
34:       Décision : Pas d’irrigation
35:       Justification : Sol lourd conserve l’humidité malgré la température → sur-
36:       veillance recommandée
37:     end if
38:   else if  $R \geq B$  then
39:     Décision : Pas d’irrigation
40:     Justification : Apports naturels suffisants post-floraison
41:   end if
42: end if
```

Suite d’algorithm 1

```
1: Stades : GRAIN LAITEUX ou PÂTEUX (JAS 135–170)
2: if  $E = \text{Grain laiteux}$  or  $E = \text{Grain pâteux}$  then
3:   if  $R < B$  and  $H < H_{seuil}$  then
4:     Décision : Irrigation
5:     Justification : Phase de remplissage — besoin élevé en eau
6:   else if  $R \geq B$  or  $H \geq H_{seuil}$  then
7:     Décision : Pas d’irrigation
8:     Justification : Apports naturels couvrent les besoins
9:   end if
10:  if  $H \geq H_{seuil}$  and  $T > T_{critique}$  then
11:    if  $S = \text{léger}$  then
12:      Décision : Irrigation
13:      Justification : Température élevée + sol sableux = évaporation rapide
        — risque de stress hydrique en phase critique de remplissage
14:    else
15:      Décision : Pas d’irrigation
16:      Justification : Bonne rétention en sol lourd
        — surveillance recommandée
17:    end if
18:  end if
19: end if

20: Stade : MATURITÉ (JAS 170–180)
21: if  $E = \text{Maturité}$  then
22:   Décision : Pas d’irrigation
23:   Justification : Fin du cycle cultural
24: end if
```

2.5.3 Sorties du système de décision

Une fois les paramètres d’entrée analysés et traités par l’algorithme, le système génère un ensemble de sorties destinées à orienter les décisions d’irrigation.

Les principales sorties sont les suivantes :

- **Décision d’irrigation** : Le système indique s’il est nécessaire ou non d’irriguer, en tenant compte des conditions météorologiques, de l’humidité du sol, ainsi que du stade de croissance de la plante.
- **Dose d’eau recommandée** : Lorsque l’irrigation est jugée nécessaire, une quantité précise d’eau est proposée selon le stade de développement du blé. Ces doses sont établies sur la base des recommandations techniques de l’ITGC.

Stade	Dose (mm)
Germination	10
Levée	10
Tallage	20
Montaison	40
Épiaison	40
Floraison	40
Grain laiteux	40
Grain pâteux	20
Maturité	0

TABLEAU 2.4 – Dose d’irrigation recommandée selon le stade phénologique[46].

- **Fréquence d’irrigation** : En plus de la dose, le système peut suggérer une fréquence d’irrigation adaptée au type de sol. Elle permet de planifier les apports dans les stades à besoin continu (tallage, montaison, remplissage) tout en évitant les excès.

Type de sol	Fréquence des irrigations	Nature de la capacité de rétention
Sol lourd	1 à 2	Forte
Sol léger	3 à 4	Faible

TABLEAU 2.5 – Fréquence d’irrigation selon le type de sol et sa capacité de rétention [46].

- **Alertes et notifications** : Si des situations inhabituelles sont détectées, comme un niveau PH de sol hors la plage optimal , un capteur le système émet une alerte pour avertir l’utilisateur.

L’ensemble de ces sorties peut être utilisé soit pour déclencher automatiquement l’irrigation via un système connecté, soit pour assister un agriculteur dans sa prise de décision.

2.6 Conclusion

Ce chapitre a présenté la conception générale du système intelligent d’aide à la décision pour l’irrigation. Il a couvert les étapes principales : la selection et le traitement du jeu de données, la construction des modèles de prédiction , la définition des variables d’entrée, ainsi que la mise en place des règles de décision. L’objectif étant de développer un système automatisé, fiable et adapté aux besoins réels en irrigation.

Chapitre 3

Implémentation et test du système

3.1 Introduction

Comme nous avons exploré la conception de notre solution dans le chapitre précédent, dans cette section, nous détaillerons les outils logiciels et les langages que nous avons utilisés, implémenterons notre application Web et conclurons par une série de tests pour vérifier l'efficacité et la fonctionnalité de notre système.

3.2 Environnement de développement

3.2.1 Outils de programmation

- **Visual Studio Code** : Visual Studio Code (VSCode), un éditeur de code source et environnement de développement intégré (IDE). C'est un logiciel open-source et multiplateforme, ce qui signifie qu'il fonctionne sur Windows, Linux et Mac. Initialement prévu pour les développeurs web, il supporte également une multitude d'autres langages de programmation comme C++, Python, Java, et bien d'autres [48].

- **MongoDB Compass** : MongoDB Compass est une interface graphique puissante permettant d'interroger, d'agréger et d'analyser vos données MongoDB dans un environnement visuel. Compass est gratuit et disponible en code source, et peut être exécuté sur macOS, Windows et Linux [49].

- **Jupyter Notebook** : Jupyter Notebook est une application web libre qui autorise la création et le partage de documents. C'est un environnement de calcul interactif web qui permet aux utilisateurs Python de concevoir des documents notebooks. Ce dispositif possède une multitude d'applications : épuration et conversion des données, simulation numérique, modélisation statistique, apprentissage automatique [10].

3.2.2 Back-end

- **Python** : Python est le langage de codage open source le plus utilisé par les spécialistes en informatique. Ce langage a pris la tête dans les domaines de la gestion d'infrastructure, de l'analyse de données et du développement de logiciels. Effectivement, l'un des atouts de Python est qu'il permet aux développeurs de focaliser leurs efforts sur leur travail plutôt que sur la méthode d'exécution de ce dernier [50].

- **FastAPI** : FastAPI est un framework web performant, de type open source, qui facilite la création d'APIs avec Python. Ce framework est aussi très rapide à coder, offrant une accélération du développement. Conçu pour être simple à apprendre et à utiliser et pour améliorer l'expérience de développement [10].

- **MongoDB** : MongoDB est un logiciel de gestion de bases de données NoSQL open source. NoSQL (et pas seulement SQL) est utilisé comme alternative aux bases de données relationnelles traditionnelles. Il est utilisé pour le stockage de données volumineuses, permettant aux organisations de stocker de grandes quantités de données tout en restant performantes [51].

3.2.3 Front-end

- **JavaScript** : JavaScript est un langage de programmation, spécifiquement un langage de script basé sur les objets. On le trouve surtout sur les pages web. Il offre, entre autres, la possibilité d'intégrer sur une page web ou HTML des animations ou des effets mineurs [50].

- **React** : React.js, souvent désignée simplement sous le nom de React, est une librairie JavaScript employée pour l'élaboration d'interfaces utilisateur. Chaque application web React est construite à partir de composants réutilisables qui forment des éléments de l'interface utilisateur. La syntaxe utilisée pour construire des applications React est appelée JSX [52].

- **Tailwind CSS** : Tailwind CSS est un framework de style utilitaire CSS (Cascading Style Sheets) qui propose des classes prédéfinies, vous permettant de créer et d'élaborer des pages web directement dans votre balisage. Cela vous donne la possibilité d'intégrer du CSS dans HTML en utilisant des classes préconfigurées [52].

3.2.4 Bibliothèques utilisées

- **TensorFlow** : TensorFlow est une bibliothèque logicielle open source dédiée au calcul numérique haute performance. Son architecture flexible facilite le déploiement des calculs sur diverses plateformes, des ordinateurs de bureau aux clusters de serveurs, en passant par les appareils mobiles et périphériques [53].

- **Scikit-learn** : Scikit-learn est une bibliothèque Python open source qui facilite l'accès au machine learning. Elle offre une interface simple et cohérente pour diverses tâches telles que la classification, la régression, le clustering, le prétraitement des données et l'évaluation de modèles [54].

- **Pandas** : est un outil d'analyse et de manipulation de données open source, rapide, puissant, flexible et facile à utiliser, basé sur le langage de programmation Python[55].

- **Numpy** : NumPy est une bibliothèque Python signifie Numerical Python. Permettant de travailler avec des tableaux. Elle propose également des fonctions pour travailler dans le domaine de l'algèbre linéaire, de la transformée de Fourier et des matrices[56]. -**Matplotlib** : Matplotlib est une bibliothèque de visualisation open source pour le langage de programmation Python, largement utilisée pour créer des graphiques statiques, animés et interactifs. Elle fournit une API orientée objet pour intégrer des graphiques dans des applications[54].

- **Seaborn** : Seaborn est une bibliothèque de visualisation exceptionnelle pour la création de graphiques statistiques en Python. Elle propose de superbes styles et palettes de couleurs par défaut pour des tracés statistiques plus attrayants [54].

- **PyMongo** : La distribution PyMongo contient des outils permettant d'interagir avec la base de données MongoDB depuis Python. Le paquet bson est une implémentation du format BSON pour Python. Le paquet pymongo est un pilote Python natif pour MongoDB[57].

- **Joblib** : Joblib est une bibliothèque open-source pour le langage Python qui simplifie la parallélisation, le cache de résultats et la répartition des tâches [10].

- **collections** : Le module Collection de Python propose différents types de conteneurs. Un conteneur est un objet permettant de stocker différents objets et d'y accéder et de les parcourir. Parmi les conteneurs intégrés, on trouve les tuples, les listes, les dictionnaires [54].

- **Pydantic** : Pydantic est une bibliothèque de validation et de gestion des paramètres de données qui exploite les annotations de type Python pour fournir des outils puissants et faciles à utiliser afin de garantir le bon format de nos données [54].

- **Random** : Le module Python Random génère des nombres aléatoires en Python. Ces nombres sont pseudo-aléatoires. Ce module permet d'effectuer des actions aléatoires, telles que la génération de nombres aléatoires, l'affichage aléatoire d'une valeur pour une liste ou une chaîne[54].

-**DateTime** : En Python, la date et l'heure ne sont pas des types de données distincts, mais un module nommé DateTime peut être importé pour gérer la date et l'heure. Le module DateTime est intégré à Python ; il n'est donc pas nécessaire de l'installer en externe [54].

3.3 Cas d'étude

Ce travail porte sur le développement d'un système intelligent d'aide à la décision pour l'irrigation. La culture du blé a été choisie comme cas d'étude, en se concentrant sur la région de la Mitidja, située dans la capitale algérienne.

3.3.1 Prétraitement des Données

3.3.1.1 Enrichissement des données

Pour garantir l'exactitude de nos prévisions météorologiques, nous avons remarqué que certaines variables essentielles manquaient dans notre ensemble de données. Afin de compléter ces informations manquantes, nous avons intégré des données issues du site Open-Meteo.com. Parmi les étapes essentielles de processus de fusion :

Les données horaires initiales ont été regroupées par date en calculant la moyenne quotidienne pour chaque variable, dans le but d'atténuer les variations et de structurer l'information sur une base quotidienne.

```
[10]: # Regrouper Les données par jour en prenant La moyenne
df_daily = df.groupby(df["datetime"].dt.date).mean()

[11]: # Renommer L'index en "date" et réinitialiser
df_daily.index.name = "date"
df_daily.reset_index(inplace=True)

# Afficher Les premières lignes du dataset journalier
df_daily.head()
```

FIGURE 3.1 – Pseudo-code pour le Regroupement des données par jour

La colonne de date a été renommée et convertie au bon format afin d'assurer une correspondance avec le second dataset

```
[20]: # Renommer La colonne 'time' en 'date' pour correspondre à df_daily
df_main.rename(columns={"time": "date"}, inplace=True)
df_main.head()

<div> ● ● ●

[21]: # Convertir 'date' en format datetime dans Les deux datasets
df_main["date"] = pd.to_datetime(df_main["date"])
df_daily["date"] = pd.to_datetime(df_daily["date"])
```

FIGURE 3.2 – Pseudo-code pour Convertir la colonne 'date' au bon format

Une jointure interne a été réalisée en se basant sur la colonne de date. Cette fusion de type interne a permis de ne conserver que les dates communes aux deux jeux de données assurant de cette manière une consistance temporelle dans les enregistrements.

```
[22]: # Fusionner Les datasets sur La colonne 'date' (jointure interne pour garder Les dates communes)
df_merged = pd.merge(df_main, df_daily, on="date", how="inner")
```

FIGURE 3.3 – Pseudo-code pour fusionner les deux jeux de données

3.3.1.2 Nettoyage des données

Cette partie est réalisée pour garantir la qualité de notre ensemble de données donc pour cela nous avons effectué un nettoyage minutieux . Parmi les 22 caractéristiques, "sunrise" et "sunset" et "precipitation-hours (h)" ont été supprimées en raison de leur inutilité et de leur non pertinence dans la prédiction, et les colonnes "rain-sum (mm)", "snowfall-sum (cm)" n'ont pas été considérées, car elles sont déjà stockées dans la variable cible "precipitations-sum (mm)", qui représente la somme totale des précipitations (rain + snow).

```
# Sélectionne que les colonnes pertinentes et utiles
features = [
    'relative_humidity_2m (%)', 'windgusts_10m_max (km/h)', 'windspeed_10m_max (km/h)', 'weathercode (wmo code)',
    'winddirection_10m_dominant (°)', 'dew_point_2m (°C)', 'pressure_msl (hPa)',
    'temperature_2m_min (°C)', 'apparent_temperature_min (°C)', 'temperature_2m_mean (°C)',
    'apparent_temperature_mean (°C)', 'apparent_temperature_max (°C)', 'temperature_2m_max (°C)',
    'et0_fao_evapotranspiration (mm)', 'shortwave_radiation_sum (MJ/m²)'
]
target = 'precipitation_sum (mm)'
```

FIGURE 3.4 – Pseudo-code pour sélectionner les variable pertinente et ignoré les inutiles

Par la suite, nous avons établi un processus dédié à l'identification et à la gestion des valeurs extrêmes (outliers). Dans l'intention de minimiser l'asymétrie de sa distribution qui pose des phénomènes typiques lors de l'entraînement du modèle du ML et plus précisément des RNN comme le LSTM, et de rectifier cette disparité et d'assurer une distribution plus équilibrée, une transformation logarithmique a été effectuée sur la variable cible précipitation.

```
#la transformation logarithmique log1p
donnees[target] = np.log1p(donnees[target])
```

FIGURE 3.5 – Pseudo-code pour la transformation logarithmique log1p

- Détection et traitement des outliers :

```
# Pipeline de détection et traitement des outliers
def detecter_outliers_iqr(series):
    Q1 = series.quantile(0.25) # Premier quartile
    Q3 = series.quantile(0.75) # Troisième quartile
    IQR = Q3 - Q1 # Intervalle interquartile
    return Q1 - 1.5 * IQR, Q3 + 1.5 * IQR # Bornes inférieure et supérieure

def interpoler_et_clipper(df, col):
    borne_inf, borne_sup = detecter_outliers_iqr(df[col]) # 1. Détection des outliers
    df.loc[(df[col] < borne_inf) | (df[col] > borne_sup), col] = np.nan # 2. Remplacement des outliers par NaN
    df[col] = df[col].interpolate(method='time') # 3. Interpolation temporelle pour remplir les NaN (valeurs aberrantes)
    borne_inf_post, borne_sup_post = detecter_outliers_iqr(df[col]) # 4. Recalcul des bornes après interpolation
    df[col] = df[col].clip(lower=borne_inf_post, upper=borne_sup_post)
    return
```

FIGURE 3.6 – Pseudo-code pour le traitement des valeurs aberrantes

Pour garantir l'exhaustivité des données , une inspection systématique a été réalisée pour identifier toute valeur manquantes.Cette étape est nécessaire car la présence de valeurs man-

quantes peut impacter négativement les performances d'apprentissage du modèle. Dans notre cas, aucune valeur manquante n'a été détectée.

```
df.apply(pd.isnull).sum()

weathercode (wmo code)          0
temperature_2m_max (°C)         0
temperature_2m_min (°C)         0
temperature_2m_mean (°C)        0
apparent_temperature_max (°C)   0
apparent_temperature_min (°C)   0
apparent_temperature_mean (°C)  0
sunrise (iso8601)               0
sunset (iso8601)                0
precipitation_sum (mm)          0
rain_sum (mm)                   0
snowfall_sum (cm)               0
precipitation_hours (h)         0
windspeed_10m_max (km/h)        0
windgusts_10m_max (km/h)        0
winddirection_10m_dominant (°)  0
shortwave_radiation_sum (MJ/m²) 0
et0_fao_evapotranspiration (mm) 0
relative_humidity_2m (%)         0
dew_point_2m (°C)               0
pressure_msl (hPa)              0
dtype: int64
```

FIGURE 3.7 – Pseudo-code pour détecter les valeurs manquantes

3.3.2 Construction du Modèle de Prévision des Précipitations

Nous avons développé un modèle de prévision des précipitations basé sur une architecture LSTM profonde. Pour cela, nous avons divisé les données en trois ensembles : **70% pour l'ensemble d'entraînement**, utilisé pour ajuster les paramètres du modèle ; **15% pour l'ensemble de validation**, permettant de régler les hyperparamètres et de prévenir le surapprentissage ; et **15% pour l'ensemble de test**, destiné à évaluer sa capacité de généralisation sur de nouvelles données.

Nous avons conçu un modèle LSTM composé de quatre couches successives avec respectivement **256, 128, 64 et 32** neurones. Les trois premières couches sont configurées avec `return_sequences=True` afin de transmettre l'ensemble des sorties à la couche suivante, tandis que la dernière couche LSTM génère une sortie finale. Des couches Dropout ont été

intégrées après certaines couches LSTM avec des taux de **30 %**, **0 %** et **20 %** pour limiter le surapprentissage. Le modèle se termine par une couche Dense à 7 neurones, permettant de prédire les précipitations sur une période de 7 jours. Pour l'entraînement, nous avons utilisé l'optimiseur Adam et la fonction de perte Huber, bien adaptée aux données météorologiques souvent bruitées.

```
[11]: from tensorflow.keras.models import Sequential

model = Sequential([
    LSTM(256, return_sequences=True, input_shape=(timesteps_in, len(features))),
    Dropout(0.3),
    LSTM(128, return_sequences=True),
    Dropout(0.),
    LSTM(64, return_sequences=True),
    Dropout(0.2),
    LSTM(32, # Pas de return_sequences ici
    Dense(7) # Sortie directe : 7 jours
])
model.compile(optimizer=Adam(learning_rate=0.001), loss=Huber())
model.summary()
```

FIGURE 3.8 – Pseudo-code du Modèle de Prédiction des Précipitations

Nous avons utilisé un entraînement régulé par des callbacks pour améliorer la performance du modèle LSTM. Ces mécanismes permettent d'éviter le surapprentissage, d'ajuster dynamiquement le taux d'apprentissage et de sauvegarder le meilleur modèle. L'apprentissage s'est déroulé sur 100 époques avec validation continue. Le modèle final a été sauvegardé pour permettre son exploitation ultérieure, notamment pour la prédiction ou l'intégration dans un système .

```
[12]: from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint

# 🍪 Callbacks pour éviter l'overfitting et sauvegarder le meilleur modèle
callbacks = [
    EarlyStopping(patience=30, restore_best_weights=True),
    ReduceLROnPlateau(patience=10, factor=0.5),
    ModelCheckpoint('model/lstm_best.keras', save_best_only=True)
]

# 🚀 Entraînement avec sauvegarde
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100,
    batch_size=32,
    callbacks=callbacks,
    verbose=1
)
```

FIGURE 3.9 – Pseudo-code pour l'Entraînement et Sauvegarde du Modèle LSTM avec Régulation Automatique

3.3.3 Les interfaces

3.3.3.1 Page de Surveillance météorologique et précipitations

En premier lieu, l'agriculteur a accès à cette interface, qui lui fournit les conditions météorologiques en cours captées par des capteurs IoT.

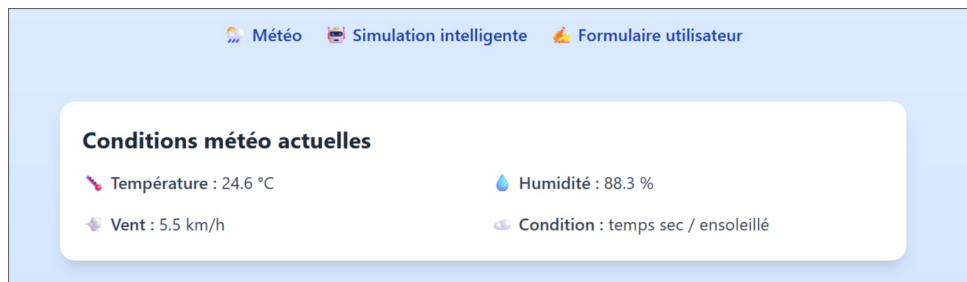


FIGURE 3.10 – les conditions météorologique captées

Il peut également consulter les prévisions de précipitations sur 7 jours .

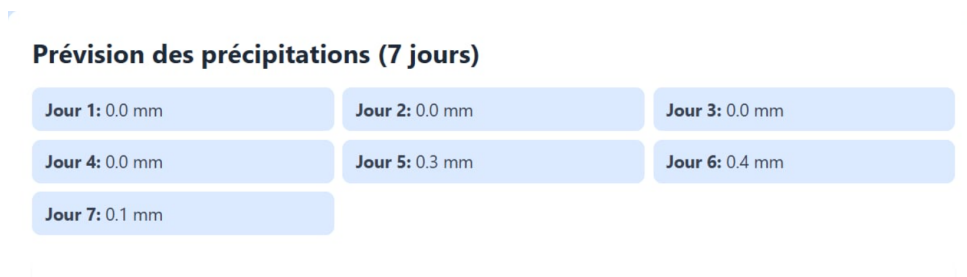


FIGURE 3.11 – les prévisions de précipitations sur 7 jours

Une représentation graphique a été réalisée afin d'illustrer l'évolution des précipitations et leurs quantités enregistrées sur une période de sept jours consécutifs.

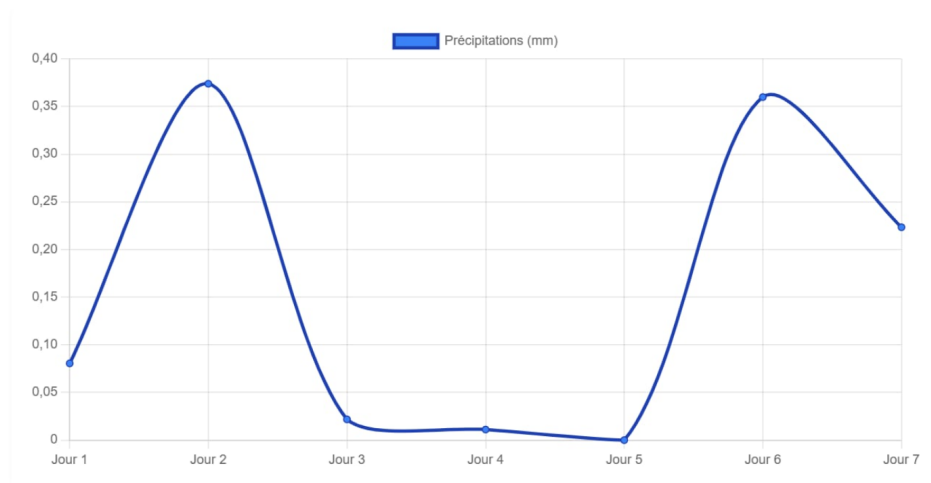


FIGURE 3.12 – Graphique des précipitations prévues

3.3.3.2 Formulaire de données climatiques et du sol

cette interface offre à l'utilisateur la possibilité d'entrer manuellement les informations météorologiques et les conditions du sol . Elle sert à alimenter le système pour simuler l'irrigation de manière personnalisée.

Formulaire manuel d'irrigation

Jours après semis: 7

pH du sol: 6,7

Type de sol: Lourd

Humidité du sol (%): 20,5

Température max (°C): 27

Précipitations prévues (7 jours)

0	0	0	0
0	0	0	0

Prédire l'irrigation

FIGURE 3.13 – formulaire utilisateur pour les données climatiques et du sol

3.3.3.3 Page de Simulation intelligente d'irrigation

Cette interface est accessible pour l'agriculteur directement depuis l'interface de suivi météorologique et des précipitations. Elle affiche également les différentes conditions du sol, telles que l'humidité, la température et propose une simulation automatisée de l'irrigation, basée sur la combinaison des données du sol provenant de capteurs IoT et des prévisions météorologiques.

Simulation automatique de l'irrigation

Température max : 29.4 °C

Humidité du sol : 48.0 %

pH du sol : 6.24

Type de sol : léger

Jours après semis : 165

FIGURE 3.14 – Simulation automatique de l'irrigation

L'interface fournit une estimation détaillée des besoins en irrigation sur les 7 jours à venir précisant la durée, la fréquence, la dose d'irrigation, ainsi que le stade de croissance de la culture



FIGURE 3.15 – Résultat d'irrigation sur 7 jours

3.3.4 Validation du système

Deux phase sont proposées pour la validation de notre système :

- La validation en consultant l'expert du domaine.
- La validation par les mesures de performance .

3.3.4.1 La validation en consultant l'expert du domaine.

Une fois le système intelligent d'aide à la décision pour l'irrigation développé, sa validation a été réalisée en collaboration avec Mr. ABBAD Mohamed , professeur en sciences agronomiques. Pour ce faire, un échantillon de parcelles de blé a été sélectionné, représentant différents stades de croissance de la culture ainsi que des conditions de sol variées. Chaque situation a été évaluée à la fois par le modèle proposé et par l'expertise de Mr. ABBADE, afin de comparer les décisions d'irrigation et d'en apprécier la cohérence.

Le tableau ci-dessous présente différents cas de test, comprenant les données d'entrée, la décision générée par le système, ainsi que celle émise par l'expert. Pour l'ensemble des situations observées, les décisions issues du système concordent avec celles de l'expert .

Cas testé	Données d'entrée	Résultats d'irrigation du système	Décision de l'expert
Cas 1	Stade : levée [10–25] Humidité du sol : 55% Température : 18 Précipitation : 0 mm (toutes les 7 jours) Jour après semis : 10 pH du sol : 5,7 Type de sol : léger	Décision : pas d'irrigation toutes les 7 jours Dose : 0 mm Durée : 0 h Justification : stade non critique ; sol humide (55%)	Décision : pas d'irrigation toutes les 7 jours Justification : stade non critique ; sol humide (55%)

Cas testé	Données d'entrée	Résultats d'irrigation du système	Décision de l'expert
Cas 2	Stade : Maturité [170–180] Humidité du sol : 40% Température max : 27°C Précipitation : 0 mm (toutes les 7 jours) Jour après semis : 170 pH du sol : 5,7 Type de sol : léger	Décision : pas d'irrigation toutes les 7 jours Dose : 0 mm Durée : 0 h Justification : fin du cycle ; pas d'irrigation nécessaire	Décision : pas d'irrigation toutes les 7 jours Justification : fin du cycle ; pas d'irrigation nécessaire
Cas 3	Stade : Floraison [125–135] Humidité du sol : 41% Température max : 24°C Précipitation : 0 mm (toutes les 7 jours) Jour après semis : 126 pH du sol : 5,7 Type de sol : lourd	Décision : pas d'irrigation toutes les 7 jours Dose : 0 mm Durée : 0 h Justification : interdiction d'irriguer durant la sortie des étamines (JAS 125–132)	Décision : pas d'irrigation toutes les 7 jours Justification : interdiction d'irriguer durant la sortie des étamines (JAS 125–132)

TABLEAU 3.1 – Tableau de validation du système d'irrigation par comparaison aux décisions de l'expert

3.3.4.2 La validation par les mesures de performance

1. Le Coefficient de détermination (R^2) :

Le coefficient de détermination (R^2) mesure la capacité d'un modèle statistique à prédire un résultat. Ce résultat est représenté par la variable dépendante du modèle. La valeur minimale possible de R^2 est 0 et la valeur maximale est 1. En d'autres termes, plus un modèle est performant dans ses prédictions, plus son R^2 sera proche de 1. Plus techniquement, le R^2 est une mesure de la qualité de l'ajustement. Il s'agit de la proportion de variance de la variable dépendante expliquée par le modèle. [58]

La formule utilisée pour calculer le coefficient de détermination (R^2) est la suivante : [58]

$$R^2 = 1 - \frac{RSS}{TSS} \quad (3.1)$$

Où :

RSS = somme des carrés des résidus

TSS = somme totale des carrés

2. Erreur absolue moyenne (MAE – Mean Absolute Error)

L'erreur absolue moyenne (MAE) est une mesure largement utilisée en statistiques et l'analyse des données qui quantifie l'ampleur moyenne des erreurs dans un ensemble de prédictions, sans tenir compte de leur direction. Elle est définie comme la moyenne des différences absolues entre les valeurs prédites et les valeurs réelles. L'analyse MAE fournit une interprétation simple de la précision des prédictions, ce qui en fait un outil essentiel pour les data scientists et les analystes lors de l'évaluation des performances des modèles de régression. En se concentrant uniquement sur la taille des erreurs, l'analyse MAE permet d'identifier l'écart entre les prédictions et les résultats réels, ce qui permet une compréhension plus claire des performances du modèle. [59]

La formule utilisée pour calculer l'erreur absolue moyenne (MAE) est : [59]

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.2)$$

Où :

- n : nombre total d'observations (ou de prédictions)
- y_i : valeur réelle (ou observée) pour l'élément i
- $\hat{y}_i$: valeur prédite (ou estimée) pour l'élément i
- $|y_i - \hat{y}_i|$: erreur absolue pour l'élément i (différence en valeur absolue entre la vraie valeur et la prédiction)

3. Erreur quadratique moyenne (MSE – Mean Squared Error)

L'erreur quadratique moyenne est un indicateur de vérification de la fiabilité d'un modèle. Cet outil étudie les écarts entre les valeurs réellement observées et les valeurs prédites par le modèle.

L'erreur quadratique est une valeur toujours positive. Plus les valeurs obtenues avec le modèle sont proches des valeurs observées, plus les écarts sont faibles et l'erreur quadratique proche de zéro. [60]

La formule de l'erreur quadratique moyenne (MSE) est la suivante : [60]

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.3)$$

Où :

- n : nombre total d'observations
- y_i : valeur observée à l'instant i
- $\hat{y}_i$: valeur prédite par le modèle à l'instant i

Nous effectuons une comparaison des performances entre quatre algorithmes : SVM, Random Forest, LSTM simple et LSTM Profond, en utilisant les indicateurs MSE, MAE et le score R^2 .

Le tableau ci-dessous présente les différentes valeurs obtenues pour chacun de ces algorithmes.

Modèle	MSE	MAE	R ²
SVR	0.0097	0.0831	0.6316
Random Forest	0.0073	0.0425	0.7223
LSTM simple	0.0044	0.0406	0.7648
LSTM Profond	0.0007	0.0139	0.9609

TABLEAU 3.2 – Tableau comparatif des performances des modèles de Prédiction des précipitations

Les histogrammes suivants illustrent la comparaison des mesures de performance des quatre algorithmes étudiés.

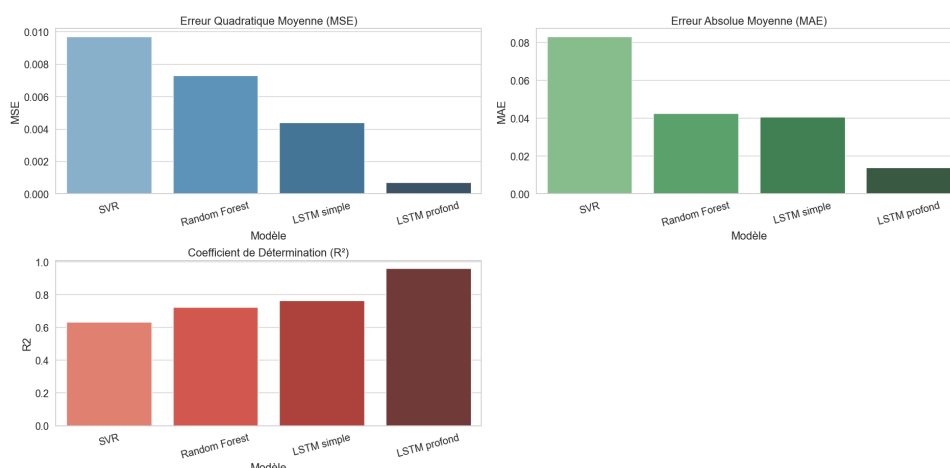


FIGURE 3.16 – Analyse comparative des performances des modèles selon les métriques MAE, MSE et R²

3.3.5 Discussion

3.3.5.1 Analyse des résultats obtenus

- **SVR** : affiche les résultats les plus faibles, avec un MSE de 0.0097, un MAE de 0.0831 et un R² de 0.63. Sa difficulté à exploiter la dimension temporelle des données limite sa performance dans ce type de tâche.
- **Random Forest** : améliore les scores (MSE = 0.0073, MAE = 0.0425, R² = 0.72), notamment grâce à sa capacité à modéliser des relations non linéaires. Toutefois, l'absence de prise en compte de la chronologie nuit à sa précision dans des séries temporelles comme les données météorologiques.
- **LSTM simple** : (MSE = 0.0044, MAE = 0.0406, R² = 0.76) tire parti de la nature séquentielle des données et offre de meilleures performances. Néanmoins, sa structure reste limitée pour détecter des variations complexes sur plusieurs jours.

- **LSTM profond** : se distingue nettement, avec un MSE très bas de 0.0007, un MAE de 0.0139 et un R^2 de 0.96. Grâce à sa profondeur, il capture efficacement les dynamiques climatiques sur une semaine complète, ce qui en fait le modèle le plus adapté dans notre cas.

3.3.5.2 Comparaison des approches algorithmiques pour la prévision des précipitations

Afin d'identifier le modèle le plus adapté à la prévision des précipitations, plusieurs approches algorithmiques ont été mises en œuvre et évaluées. Ces méthodes incluent des modèles classiques d'apprentissage automatique tels que le SVM et le Random Forest, ainsi que des modèles d'apprentissage profond comme le LSTM simple et le LSTM profond.

Cette section présente une analyse comparative de leurs performances en termes de précision, de capacité à capturer la dynamique temporelle, et de robustesse face à la complexité des données météorologiques. L'objectif est de justifier le choix final de l'algorithme le plus performant et le mieux adapté au contexte de l'agriculture de précision.

SVR	Random Forest	LSTM simple	LSTM profond
Ne modélise pas les séquences temporelles	Approche statique	Capable de gérer des séquences simples	Capte les dépendances temporelles longues et complexes
Précision faible sur séries météo non linéaires	Précision moyenne sur séries météo non linéaires	Bonne précision sur séries météo non linéaires	Très haute précision sur données non linéaires
Limité aux relations simples	Ne capture pas les interactions temporelles	Capacité limitée à long terme	Apprentissage hiérarchique grâce aux couches profondes
Sensible à l'overfitting	Stable mais peu évolutif	Performances dégradées sur gros volumes	Exploite pleinement les grands volumes de données
Non adapté aux données séquentielles réelles	Non adapté aux données séquentielles réelles	Bonne sur séries simples	Très bonne pour les séries longues, bruitées, irrégulières

TABLEAU 3.3 – Tableau comparatif des approches prédictives

3.3.5.3 Tests expérimentaux et choix de l'architecture LSTM

L'objectif de cette section est de présenter les différents tests réalisés pour optimiser l'architecture du modèle LSTM utilisé dans la prévision météorologique, en justifiant progressivement l'évolution vers le modèle final **LSTM profond**.

— Test 1 : LSTM simple — Architecture légère

Dans ce premier test, une architecture LSTM simple composée de deux couches (128 et 64 unités) a été mise en œuvre avec une couche de dropout de 0.3 pour limiter le surapprentissage. Le modèle a été entraîné sur les données météorologiques pour prédire les précipitations sur 7 jours. Les performances obtenues montrent une précision correcte avec une **erreur quadratique moyenne (MSE)** de **0.0044**, une **erreur absolue moyenne (MAE)** de **0.0406**, et un **R²** de **0.7648**. Bien que satisfaisant pour une architecture de base, ce modèle reste limité dans sa capacité à modéliser les dépendances temporelles complexes.

```
[8]: # 🍷 Modèle LSTM simple
model = Sequential([
    LSTM(128, return_sequences=True, input_shape=(timesteps_in, len(features))),
    Dropout(0.3),
    LSTM(64), # Pas de return_sequences ici
    Dense(7) # Sortie directe : 7 jours
])
model.compile(optimizer=Adam(learning_rate=0.001), loss=Huber())
model.summary()
```

FIGURE 3.17 – pseudo-code pour le premier test du modèle LSTM.

Test 2 : LSTM plus large — plus de neurones

Dans ce second test, le nombre d'unités a été augmenté à 256 puis 128 pour renforcer la capacité d'apprentissage du modèle. Cette architecture plus large permet une meilleure représentation des données séquentielles. Les résultats s'améliorent sensiblement : **MSE = 0.0031**, **MAE = 0.0333**, **R² = 0.8366**. Le gain de performance reste modéré, et le modèle montre encore des limites lorsqu'il s'agit de capturer des schémas complexes ou du bruit dans les séries temporelles.

```
[8]: # 🍷 Modèle LSTM simple
model = Sequential([
    LSTM(256, return_sequences=True, input_shape=(timesteps_in, len(features))),
    Dropout(0.3),
    LSTM(128), # Pas de return_sequences ici
    Dense(7) # Sortie directe : 7 jours
])
model.compile(optimizer=Adam(learning_rate=0.001), loss=Huber())
model.summary()
```

FIGURE 3.18 – pseudo-code pour le deuxième test du modèle LSTM.

Test 3 : LSTM profond (3 couches)

Le troisième test introduit une architecture plus profonde, avec trois couches LSTM (256, 128, 64), entrecoupées de couches dropout. Cette approche permet d'augmenter la profondeur du réseau tout en maintenant un bon contrôle de la régularisation. Les résultats sont nettement meilleurs : **MSE = 0.0022**, **MAE = 0.0260**, **R² = 0.8822**. Le modèle commence à mieux saisir les dépendances complexes à long terme, et montre une meilleure capacité de généralisation sur des données bruitées.

```
[10]: from tensorflow.keras.models import Sequential

# 🍷 Modèle LSTM simple
model = Sequential([
    LSTM(256, return_sequences=True, input_shape=(timesteps_in, len(features))),
    Dropout(0.3),
    LSTM(128, return_sequences=True),
    Dropout(0.3),
    LSTM(64, # Pas de return_sequences ici
    Dense(7) # Sortie directe : 7 jours
])
model.compile(optimizer=Adam(learning_rate=0.001), loss=Huber())
model.summary()
```

FIGURE 3.19 – pseudo-code pour le troisieme test du modèle LSTM.

Test 4 : LSTM très profond (4 couches + affinement)

Dans ce dernier test, une architecture LSTM plus profonde est mise en place, composée de quatre couches LSTM (256 → 128 → 64 → 32) avec des couches dropout ajustées (0.3, 0.0, 0.2) et une couche finale Dense pour la prédiction des 7 jours. Ce modèle tire pleinement parti de la profondeur pour extraire les dépendances temporelles complexes dans les données météorologiques. Les performances atteignent un niveau optimal avec **MSE = 0.0007**, **MAE = 0.0139** et un **R² = 0.9609**, ce qui justifie clairement le choix de cette architecture comme modèle final.

```
[11]: from tensorflow.keras.models import Sequential

model = Sequential([
    LSTM(256, return_sequences=True, input_shape=(timesteps_in, len(features))),
    Dropout(0.3),
    LSTM(128, return_sequences=True),
    Dropout(0.),
    LSTM(64, return_sequences=True),
    Dropout(0.2),
    LSTM(32, # Pas de return_sequences ici
    Dense(7) # Sortie directe : 7 jours
])
model.compile(optimizer=Adam(learning_rate=0.001), loss=Huber())
model.summary()
```

FIGURE 3.20 – pseudo-code pour le quatrieme test du modèle LSTM.

3.3.6 Conclusion

Ce chapitre a permis de retracer l'ensemble des étapes liées à la conception et à la mise en œuvre du système d'aide à la décision pour l'irrigation. Après avoir présenté les outils et logiciels utilisés, nous avons illustré le fonctionnement du système à travers des interfaces claires. L'ensemble du processus de développement a été expliqué, depuis la structuration des modules jusqu'aux tests réalisés pour en évaluer les performances. Une étude de cas validée par M. Abbade, professeur en sciences agronomiques à l'Université de Blida 1, est venue appuyer la pertinence du système dans un contexte réel.

Conclusion et perspectives

Ce mémoire est le fruit d'un semestre de recherche intensive dans le domaine de l'agriculture, avec un intérêt particulier porté sur la prévision météorologique et la gestion intelligente de l'irrigation à l'aide des technologies de l'Internet des objets (IoT). Dans un premier temps, nous avons présenté un état de l'art détaillé, abordant les approches existantes en matière de prévision des précipitations et mettant en lumière le rôle croissant des technologies IoT, tout en soulignant leur intérêt pour les problématiques agricoles.

Par la suite, nous avons conçu et réalisé un système intelligent basé sur des algorithmes de machine learning et deep learning pour anticiper les précipitations sur une période de sept jours. Ce modèle alimente un système d'aide à la décision pour l'irrigation, structuré à partir de règles définies avec l'appui d'un expert en sciences agronomiques. Nous avons également développé une interface claire et fonctionnelle qui permet de consulter les prévisions, les données du sol et de simuler les décisions d'irrigation de façon automatisée et intuitive.

Ce projet a également permis de consolider nos compétences techniques, notamment en programmation Python, en bases de données NoSQL (MongoDB) et en développement d'API avec FastAPI, tout en ouvrant des perspectives concrètes de recherche dans le secteur agricole.

Cependant, certaines limitations ont été rencontrées, notamment l'absence de jeux de données adaptés aux conditions spécifiques de l'agriculture en Algérie. Cela souligne la nécessité de créer des bases de données locales et de sensibiliser davantage les agriculteurs à l'intégration de solutions intelligentes dans leurs pratiques. Ces aspects constituent des axes d'amélioration et d'approfondissement pour les travaux futurs.

Afin de poursuivre le développement de ce projet, certaines pistes d'amélioration et d'approfondissement ont été identifiées, tel que :

- créer ou enrichir un jeu de données algérien sur l'irrigation, actuellement peu disponible.
- Rendre le système facilement déployable dans diverses zones agricoles avec différents climats et types de sol.
- Adapter les modèles de prédiction aux besoins spécifiques de différentes plantes (ex. : maïs, tomate).

- Permettre de suivre et comparer plusieurs champs dans une seule plateforme, utile pour les grandes exploitations ou les coopératives.

Bibliographie

- [1] Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jeroen Bogaardt. Big data in smart farming – a review. *Agricultural Systems*, 153 :69–80, 2017.
- [2] V. Dhanaraju, R. Kumaravelan, P. Parthasarathy, R. Kumar, and K. Kotecha. Machine learning applications for precision agriculture : A comprehensive review. *IEEE Access*, 8 :205564–205588, 2020.
- [3] H. Moumen, A. Ait Lahcen, and H. Mousannif. Artificial intelligence in agriculture : A comprehensive review. *Materials Today : Proceedings*, 85 :551–560, 2023.
- [4] Muthumanickam Dhanaraju, Poongodi Chenniappan, Kumaraperumal Ramalingam, Selaperumal Pazhanivelan, and Ragunath Kaliaperumal. Smart farming : Internet of things (iot)-based sustainable agriculture. *Agriculture*, 12(10) :1745, 2022.
- [5] Nishant K. Sinha, Jitendra Kumar, Dhiraj Kumar, Rahul Mishra, J. K. Thakur, K. C. Shinogi, Jayaraman Somasundaram, and M. Mohanty. Application of artificial intelligence (ai) in agriculture : An indian perspective. *Harit Dhara*, 5(2) :9–11, July–December 2022.
- [6] Konstantinos G. Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. Machine learning in agriculture : A review. *Sensors*, 18(8) :2674, 2018.
- [7] Data Bird. Regression in machine learning : Definition and applications, Consulté en 2025.
- [8] Geeta University. Application of correlation and regression in agriculture.
- [9] DataCamp. Classification in machine learning : Techniques and applications.
- [10] DataScientest. Algorithme de classification : définition et principaux modèles, Consulté en 2025.
- [11] EOS Data Analytics. Classification des cultures par télédétection.
- [12] EvolutionsAI. 8 applications pratiques de l’ia en agriculture.

- [13] Google Developers. Algorithmes de clustering.
- [14] Martin Ingeli, Jana Galambošová, Renáta Benda Prokešová, and Vladimír Rataj. Application of clustering method to determine production zones of field. *Acta Technologica Agriculturae*, 2 :42–45, 2015.
- [15] David Hall, Feras Dayoub, Jason Kulk, and Chris McCool. Towards unsupervised weed scouting for agricultural robotics. *arXiv preprint arXiv :1702.01247*, 2017.
- [16] IBM. Que sont les réseaux neuronaux.
- [17] Sandesh Ramesh, Anirudh Hebbar, Varun Yadav, Thulasiram Gunta, and A. Balachandra. CYPUR-NN : Crop yield prediction using regression and neural networks. *arXiv preprint arXiv :2011.13265*, 2020.
- [18] IBM. Qu’est-ce que le deep learning.
- [19] Chaire AgroTIC. Deep learning et agriculture : Comprendre le potentiel et les défis à relever. Technical report, Chaire AgroTIC, novembre 2018. Consulté le 14 mars 2025.
- [20] Somayya Madakam, R. Ramaswamy, and Siddharth Tripathi. Internet of things (iot) : A literature review. *Journal of Computer and Communications*, 3(5) :164–173, 2015.
- [21] Mutualia. L’internet des objets (iot) pour une agriculture plus intelligente, 2021. Consulté le 11 février 2025.
- [22] Radouan Ait Mouha. Internet of things (iot). *Journal of Data Analysis and Information Processing*, 9 :77–101, 2021.
- [23] Encyclopedia MDPI. The internet of things. <https://encyclopedia.pub/entry/2732>, 2020.
- [24] Kamel Bestaali and Salim Benmesbah. Modélisation des connaissances incertaines dans l’internet des objet (iot) : contexte des objets connectés médicaux. Mémoire de master, Université de Blida, 2022.
- [25] IoT Industriel. Architecture iot : l’essentiel à savoir, 2024. Consulté le 13 février 2025.
- [26] Carlos Barraza. Avantages et inconvénients de l’internet des objets, 2024.
- [27] The Engineering Projects. Introduction to dht22, 2019.
- [28] Decagon Devices. Gs3 sensor : Ruggedized soil moisture, temperature, & electrical conductivity sensor. Document PDF, disponible en ligne.
- [29] Semageek. Capteur de température et humidité de base dht11 et extras. Boutique en ligne Semageek.

- [30] Davis Instruments. Davis 6332 – anemometer transmitter kit.
- [31] SSDIELECT ELECTRONICA SAS. Module capteur détecteur de pluie yl-83.
- [32] Apogee Instruments. Sp-110-ss : Pyranomètre autonome.
- [33] Rui Santos. Guide for soil moisture sensor yl-69 or hl-69 with arduino.
- [34] Milton Kanda and Florence Mutua. Spatio-temporal prediction of rainfall and runoff in data-scarce basins using lstm and wavelet neural networks. *Complex & Intelligent Systems*, 8 :1503–1520, 2022.
- [35] Nitin Singh, Saurabh Chaturvedi, and Shamim Akhter. Weather forecasting using machine learning algorithm. In *2019 IEEE International Conference on Innovations in Science, Engineering and Technology (ICISSET)*, pages 171–174. IEEE, 2019.
- [36] Baoxiang Pan, Kuolin Hsu, Amir AghaKouchak, and Soroosh Sorooshian. Improving precipitation estimation using convolutional neural network. *Water Resources Research*, 55(3) :2301–2321, 2019.
- [37] Youness Tace, Mohamed Tabaa, Sanaa Elfilali, Cherkaoui Leghris, Hassna Bensag, and Eric Renault. Smart irrigation system based on iot and machine learning. *Energy Reports*, 8 :1025–1036, 2022.
- [38] Mohammed Islam Khellouf and Mohammed Amine Tebaa. Étude et réalisation d’un système d’irrigation intelligent, 2021.
- [39] Open-Meteo. Open-meteo : Weather api for non-commercial use. <https://open-meteo.com/>.
- [40] Merwan Bekkar. Algiers weather data (2002 - 2023). <https://www.kaggle.com/datasets/bekkarmerwan/algers-weather-data-2002-2023>, 2023.
- [41] Blent.ai. Random forest : Comment ça marche ?, 2023.
- [42] Nandini Verma. An introduction to support vector regression (svr) in machine learning, 2023.
- [43] Slimane Bouhouia and Omar Zaouak. Reconnaissance des caractères manuscrits arabe par réseau de neurones récurrent (lstm). Mémoire de fin d’Études – master en informatique, option systèmes intelligents, Université Ahmed Draia - Adrar, Adrar, Algérie, 2023. <https://dspace.univ-adrar.edu.dz/jspui/handle/123456789/8738>.
- [44] Skilldux. Deep lstm : Architecture and overview, 2023. Consulté le 16 juin 2025.
- [45] Stackademic. Complete guide to learn lstm models – types, applications, and when to use which model, 2023.

- [46] Institut Technique de Gestion des Cultures (ITGC). L'irrigation d'appoint du blé. Technical report, ITGC, Algérie, 2020. Fiche technique, version optimisée compressée.
- [47] J.C. Zadoks, T.T. Chang, and C.F. Konzak. A decimal code for the growth stages of cereals. *Weed Research*, 14(6) :415–421, 1974.
- [48] Bility.fr. Définition visual studio code.
- [49] MongoDB, Inc. Mongodb compass documentation.
- [50] Journal du Net. Python : définition et utilisation de ce langage informatique.
- [51] TechTarget. Mongodb.
- [52] Kinsta. Qu'est-ce que react js ?
- [53] The TensorFlow Authors. Tensorflow. <https://pypi.org/project/tensorflow/>. Disponible sur PyPI.
- [54] GeeksforGeeks. Geeksforgeeks – a computer science portal for geeks. Consulté en juin 2025.
- [55] Pandas Development Team. Pandas : Python Data Analysis Library. <https://pandas.pydata.org/>.
- [56] W3Schools. Numpy introduction - w3schools.
- [57] Inc. Anaconda. Pymongo : : Anaconda.org.
- [58] Scribbr. Coefficient of determination (R^2) | definition & calculation, 2023.
- [59] StatisticsEasily. Qu'est-ce que c'est que l'erreur absolue, mae.
- [60] DataScientest. Erreur quadratique moyenne : définition et utilité, 2023.