

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme d'Ingénieur

Filière : Génie industriel
Spécialité : Management industriel et logistique

Présenté par :

BENCHALGO Salima
CHIKH Aymen

Thème

Visualisation des séries chronologique (time series) dans les systèmes industriels

Soutenu publiquement, le 04/07/2023, devant le jury composé de :

M. MALIKI Fouad	MCA	ESSA. Tlemcen	Président
M. OUHOUD Amina	MCB	ESSA. Tlemcen	Directeur de mémoire
M. BRAHAMI Mustapha Anwar	MCA	ESSA. Tlemcen	Examineur 1
Mme. SEKKAL Nour el houda	MAB	ESSA. Tlemcen	Examineur 2

Année universitaire : 2022 / 2023

Remerciements

Nous tenons tout d'abord à exprimer notre reconnaissance envers Dieu le Tout-Puissant, qui nous a accordé la force et la patience nécessaires pour accomplir ce modeste travail.

En second lieu, nous souhaitons prendre un moment pour exprimer notre profonde gratitude envers notre encadrante, Amina OUHOUD, pour son soutien et son encadrement tout au long de la rédaction de notre mémoire. Ton expertise, ta patience et ton dévouement ont été des éléments essentiels qui nous ont permis de mener à bien cette étape importante de notre parcours académique.

Nous tenons également à exprimer notre gratitude envers tous les autres professeurs qui ont contribué à notre formation académique, ainsi qu'aux membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'examiner ce travail.

Enfin, nous souhaitons remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail, notamment nos collègues de la classe GI. Votre soutien et votre collaboration ont été précieux.

Avec notre plus sincère gratitude,

Aymen Chikh

Salima Benchalgo

Dédicace

“

*A l'homme de ma vie, mon exemple éternel, mon soutien
moral et source de joie et de bonheur, celui qui s'est
toujours sacrifié pour me voir réussir, que dieu te garde
dans son vaste paradis, à toi mon père.*

*A ma maman qui m'a soutenu et encouragé durant ces
années d'études.*

A mes deux sœurs, et mes frères,

*Aux personnes qui m'ont toujours aidé et encouragé, qui
étaient toujours à mes côtés*

A tous ceux que j'aime

A vous

”

Salima

Table des matières

Remerciements	II
Dédicace	III
Introduction générale	2
1 Généralités sur l’IoT	4
1.1 Notions de base	5
1.1.1 Les appareils connectés	5
1.1.2 Les réseaux de capteurs	6
1.1.3 Les protocoles de communication	6
1.1.4 Les plateformes IoT	6
1.1.5 L’analyse de données	6
1.1.6 La sécurité IoT	6
1.1.7 Big data	6
1.2 Définition de l’IoT	7
1.3 Fonctionnement de l’IoT	7
1.3.1 Collecte de données	7
1.3.2 Traitement de données	8
1.3.3 Transmission de données	8
1.3.4 Analyse de données	8
1.3.5 Prise de décision	8
1.4 Les fonctions de l’objet connecté	8
1.4.1 Les capteurs	8
1.4.2 Les sources d’énergies	9
1.4.3 Les actionneurs	10
1.4.4 La connectivité	11
1.5 Les protocoles de communication dans l’IoT	11
1.5.1 Le Protocole HTTP	11
1.5.2 Le Protocole MQTT	12
1.5.3 Le Protocole CoAP	13
1.5.4 Comparaison entre HTTP, MQTT et CoAP	13
1.5.5 Les technologies de communication dans l’IoT	14
1.6 Architecture de l’IoT	15
1.6.1 Couche de perception	16
1.6.2 Couche réseau	16
1.6.3 Couche de traitement des données	16
1.6.4 Couche d’application	16

1.7	Les avantages de l’IoT	16
1.8	Les défis de l’IoT	17
1.9	Les enjeux éthiques et environnementaux de l’IoT	19
1.10	Les perspectives d’avenir de l’IoT	20
1.10.1	Évolution des technologies IoT (Intelligence Artificielle IoT)	20
1.10.2	Développement de nouvelles applications	20
1.10.3	Implications sociétales de l’IoT	21
1.11	L’état de l’art	21
2	Les séries chronologiques	25
2.1	Définition d’une donnée	26
2.2	Les types de données	27
2.2.1	Les données transversales	27
2.2.2	Données longitudinales (Longitudinal data)	27
2.2.3	Données catégoriques (Categorical data)	28
2.2.4	Données numériques (Numerical data)	28
2.2.5	Données temporelles (Time series data)	28
2.3	Définition d’une série chronologique	28
2.4	Les exigences des séries chronologiques	29
2.4.1	Régularité temporelle	29
2.4.2	Ordre temporel	29
2.4.3	Stationnarité	29
2.4.4	Absence de valeurs manquantes	29
2.4.5	Homoscédasticité	29
2.4.6	Absence de tendances saisonnières	30
2.5	Représentation graphique des données d’une série temporelle	30
2.6	Base de données des séries chronologique	30
2.6.1	Définition d’une base de données des séries chronologiques	30
2.6.2	Importance du Time Series Database (TSBD)	31
2.6.3	Différence entre les BDD temporelles et les BDD relationnelles	32
2.6.4	Les avantages de l’utilisation d’une TSDB	33
2.6.5	Les applications des séries chronologiques	34
2.6.6	Inconvénients des prévisions des séries chronologiques	35
2.6.7	Les principales TSDB	36
2.7	InfluxDB :	38
2.7.1	Définition	38
2.7.2	Principales caractéristiques	38
2.7.3	InfluxDB Cloud	39
2.7.4	Avantages et limitations d’InfluxDB :	40
2.7.5	Les clients d’InfluxDB	41
2.7.6	Architecture d’InfluxDB	41
2.7.7	les concept-clés du modèle de données d’InfluxDB	42
2.7.8	Les séries chronologiques et l’intelligence artificielle	43
2.7.9	Les méthodes de prévision d’une série chronologique	44
2.7.10	Conclusion :	45

3	Etude de cas	47
3.1	L'objectif de projet	48
3.2	Pourquoi on a choisi l'agriculture ?	49
3.3	Choix des données	50
3.3.1	Température	50
3.3.2	Humidité	50
3.3.3	Précipitations	50
3.3.4	Vitesse du vent	50
3.3.5	pH du sol	50
3.3.6	Niveau de nutriments	50
3.3.7	Niveau d'humidité du sol	50
3.3.8	pH de l'eau	51
3.3.9	Niveau de sel	51
3.3.10	Niveau de contamination	51
3.4	les outils et les technologies utilisés	51
3.4.1	InfluxDB	51
3.4.2	Mosquitto	52
3.4.3	Le protocole MQTT (Message Queue Telemetry Transport)	52
3.4.4	Framework Web Python	53
3.4.5	Les bibliothèques utilisées :	55
3.4.6	Format JSON :	57
3.5	Étapes clés de la réalisation de notre projet	57
3.6	Implémentation du projet	58
3.6.1	L'importation des modules :	59
3.6.2	Définition des variables :	59
3.6.3	Création du client MQTT :	59
3.6.4	Démarrage du service client MQTT :	59
3.6.5	Connexion au serveur MQTT:	59
3.6.6	Définition de fonctions utilitaires :	59
3.6.7	Boucle principale :	59
3.7	Fonctionnement et visualisation des données en utilisant InfluxDB :	60
3.8	Démarrage de la base de données influx db	60
3.9	Création du tableau de bord	62
3.10	Les tableaux de bord réalisés	63
3.11	Grafana	66
3.11.1	Définition	66
3.11.2	Les avantages d'utilisation de Grafana	67
3.11.3	Les inconvénients	67
3.11.4	A quoi sert Grafana ?	68
3.11.5	Démarrage et Accès à l'interface web de Grafana	68
3.11.6	Présentation de l'interface de Grafana	68
3.11.7	La connexion de grafana avec influxdb :	69
3.11.8	Création de visualisations	70
3.12	Grafana V.S InfluxDB	73
3.13	Conclusion	73

Table des matières

Conclusion générale	75
Bibliography	77

Liste des figures

1.1	Fonctionnement de l’IoT.	7
1.2	Capteur de mouvement et température	9
1.3	Piles de type Duracell	9
1.4	Capteur d’énergie	10
1.5	Lecteur RFID	10
1.6	Vérin pneumatique Alarme	10
1.7	Les technologies de connectivité.	11
1.8	Le protocole Http	12
1.9	MQTT Protocol	12
1.10	CoAP Protocol	13
1.11	Architecture de l’IoT.	15
2.1	Représentation graphique des données d’une série temporelle.	30
2.2	Classement des bases de données des séries chronologiques d’après le site web DB-Engines ranking.	37
3.1	Les commandes d’installation	51
3.2	L’architecture du protocole Mqtt	52
3.3	La commande d’installation de flask	54
3.4	La commande d’installation de flask	55
3.5	La commande d’installation de paho-mqtt	56
3.6	La commande d’installation de flask	56
3.7	La commande d’installation de dateTime	57
3.8	L’architecture globale du projet	57
3.9	Initialisation du serveur mosquitto	60
3.10	La commande d’installation de flask	61
3.11	La commande d’installation de flask	61
3.12	La commande d’installation de flask	61
3.13	La commande d’installation de flask	62
3.14	Tableau de bord des conditions météorologiques	63
3.15	Tableau de bord de température	64
3.16	Tableau de bord d’irrigation	64
3.17	Tableau de bord de pH du sol	64
3.18	Tableau de bord de la qualité de l’eau pour la croissance des cultures	65
3.19	Tableau de bord de la qualité de l’eau pour la croissance des cultures	65
3.20	Une format de données d’un fichier CSV	66
3.21	Présentation de l’interface de grafana	69
3.22	Ajouter influxdb comme une source de données	69

3.23	Configuration de la source de données	70
3.24	Le test de la connexion à la source de données influxDB est réussi	70
3.25	La configuration des données de température depuis influxDB	71
3.26	Tableau de bord de températures	71
3.27	Tableau de bord des données météorologiques	71
3.28	Tableau de bord données de sol	72
3.29	Tableau de bord données sur la qualité de l'eau	72

Liste des tableaux

- 1.1 Comparaison entre les protocoles 14
- 3.1 Tableau comparative entre Grafana et InfluxDB 73

Liste des abréviations

IoT	<i>Internet of Things</i>
DB	<i>Data Base</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
CoAP	<i>Constrained Application Protocol</i>
NFC	<i>Near Field Communication</i>
IA	<i>Intelligence Artificielle</i>
IAOT	<i>Artificial Intelligence of Things</i>
TS	<i>Time Series</i>
TSDB	<i>Time Series Data Bases</i>
BDD	<i>Base de données</i>
RDB	<i>Relational Data Base</i>
SVM	<i>Support vector machines</i>
API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
GC	<i>Google Cloud</i>
ARIMA	<i>Autoregressive Integrated Moving Average</i>
RNN	<i>Réseaux de Neurones</i>

Résumé :

La technologie d'Internet of Things (IoT) a attiré beaucoup d'attention ces dernières années, en particulier dans le domaine de la supervision des données.

L'émergence de l'IoT offre un grand potentiel pour le développement de nouveaux services et applications connectant le monde physique au monde virtuel, un processus qui n'était pas possible auparavant.

L'objectif de ce travail est de mettre en œuvre une conception d'une application spécifique pour le suivi en temps réel basée sur l'IoT en utilisant des outils essentiels tels qu'InfluxDB, Grafana et MQTT.

mots clés: Internet des objets, temps réel, InfluxDB

Abstract :

The Internet of Things (IoT) technology has attracted a lot of attention in recent years, particularly in the field of data monitoring.

The emergence of IoT offers great potential for the development of new services and applications that connect the physical world to the virtual world, a process that was not possible before.

The objective of this work is to implement the design of a specific real-time tracking application based on IoT using essential tools such as InfluxDB, Grafana, and MQTT.

Keywords: Internet of Things, time Series, InfluxDB

الملخص:

جذبت أنترنت الأشياء الكثير من الاهتمام في السنوات الأخيرة، خاصة في مجال مراقبة البيانات.

تتيح ظهور هذه التكنولوجيا فرصًا كبيرة لتطوير خدمات وتطبيقات جديدة تربط العالم الفعلي بالعالم الافتراضي، وهي عملية لم تكن ممكنة في الماضي.

هدف هذا العمل هو تصميم تطبيق محدد لمتابعة البيانات في الوقت الحقيقي باستخدام أدوات أساسية مثل: InfluxDB, Grafana, MQTT.

الكلمات المفتاحية: أنترنت الأشياء, سلسلة زمنية, InfluxDB .

Introduction générale

L'émergence de l'Internet des objets (IoT) a engendré une transformation significative de notre interaction avec le monde physique en connectant divers objets à Internet. Cette avancée technologique a ouvert de nouvelles perspectives pour le développement d'applications intelligentes et innovantes. Cette étude se focalise sur la conception d'une application spécifique basée sur l'IoT en utilisant des outils essentiels tels qu'InfluxDB, Grafana et MQTT. Le présent mémoire débute par une introduction générale à l'IoT, puis aborde les concepts relatifs aux séries chronologiques avant de se plonger dans une description détaillée des étapes de simulation du projet.

Le premier chapitre de ce mémoire se concentre sur les fondements de l'IoT. Il expose les principes essentiels de cette technologie, son architecture et les acteurs majeurs qui y participent. Les avantages et les défis liés à l'utilisation de l'IoT dans divers domaines tels que la santé, l'industrie et la domotique sont également discutés. Par ailleurs, une attention particulière est portée sur l'importance des protocoles de communication dans les applications IoT, en mettant en évidence MQTT (Message Queuing Telemetry Transport), qui assure une communication fiable et légère entre les périphériques connectés.

Le deuxième chapitre se concentre sur l'analyse des séries chronologiques, qui jouent un rôle crucial dans le contexte de l'IoT. Les concepts clés tels que la tendance, la saisonnalité et les modèles de prévision sont explorés. L'analyse des données temporelles collectées par les capteurs IoT permet de détecter des schémas, de prévoir les tendances futures et de prendre des décisions éclairées. Différentes méthodes d'analyse des séries chronologiques sont présentées, notamment les modèles ARIMA et les réseaux de neurones récurrents (RNN).

Le troisième chapitre propose une description détaillée des étapes de simulation du projet. L'application IoT en cours de développement est introduite, en soulignant ses objectifs, ses fonctionnalités et son architecture. Pour stocker et gérer efficacement les données IoT, InfluxDB, une base de données spécifiquement conçue pour les séries chronologiques, est utilisée. L'utilisation de Grafana, une plateforme de visualisation des données, est également abordée, permettant ainsi la création de tableaux de bord interactifs et de graphiques en temps réel. Enfin, une explication détaillée est fournie sur l'utilisation de MQTT pour la communication entre les dispositifs IoT, assurant une transmission fiable et efficace des données.

En conclusion, ce mémoire explore le développement d'une application IoT en s'appuyant sur des concepts clés tels que l'IoT, les séries chronologiques, InfluxDB, Grafana et MQTT. Il met en évidence l'importance de ces outils et technologies dans la création d'applications IoT efficaces et évolutives. En combinant les connaissances relatives à l'IoT, aux séries chronologiques et aux étapes de simulation du projet, il devient possible de concevoir et de développer des applications IoT robustes qui exploitent pleinement le potentiel des données collectées par les dispositifs connectés.

Chapter 1

Généralités sur l'IoT

Introduction

L'Internet des objets (IoT) est une technologie majeure qui est en train de transformer le paysage connecté actuel de manière significative. Grâce à l'IoT, des milliards d'objets peuvent être connectés à Internet et interagir entre eux, permettant ainsi une collecte et une analyse de données à grande échelle. Cette avancée technologique a solidement établi l'IoT comme un pilier central de l'industrie 4.0, des villes intelligentes et des foyers connectés.

Les bénéfices offerts par l'IoT sont considérables. Il permet d'optimiser les processus industriels, d'améliorer la qualité de vie des individus et d'accroître l'efficacité énergétique. Les dispositifs connectés fournissent des données précieuses qui permettent de prendre des décisions éclairées, d'automatiser des tâches et de développer de nouvelles applications intelligentes. Cependant, l'IoT présente également des défis majeurs. La sécurité des données et des dispositifs connectés constitue une préoccupation cruciale, étant donné que la multiplication des points d'accès potentiels accroît les risques de cyberattaques. De plus, l'intégration de l'IoT avec les systèmes existants peut s'avérer complexe, nécessitant une interopérabilité et une normalisation adéquates. Enfin, les coûts d'investissement liés à l'infrastructure et aux dispositifs IoT peuvent être considérables, ce qui peut freiner leur adoption à grande échelle.

Ce chapitre vise à fournir une présentation détaillée des principes fondamentaux de l'IoT. Nous examinerons en profondeur son architecture essentielle et son mode de fonctionnement, en mettant l'accent sur la communication entre les objets connectés et l'échange de données. En outre, nous analyserons les avantages concrets que l'IoT apporte dans divers secteurs, tels que l'industrie, l'environnement et la santé. Nous n'omettrons pas les défis inhérents à l'IoT, notamment en matière de sécurité, d'intégration et de coûts. Nous aborderons également les enjeux éthiques et environnementaux découlant de cette technologie en constante évolution. Enfin, nous discuterons des perspectives futures de l'IoT, en mettant en lumière les tendances émergentes et les développements à venir.

1.1 Notions de base

L'avènement de l'Internet des objets (IoT) a donné naissance à une multitude d'appareils connectés qui transforment notre environnement quotidien. Ces appareils, dotés de capteurs, de processeurs, de communications sans fil et de logiciels, ont la capacité de collecter, traiter et transmettre des données. Ils représentent une nouvelle génération d'objets physiques intelligents qui interagissent avec leur environnement et élargissent les possibilités de l'IoT.

1.1.1 Les appareils connectés

Ce sont des objets physiques qui sont équipés de capteurs, de processeurs, de communications sans fil et de logiciels pour collecter, traiter et transmettre des données.

1.1.2 Les réseaux de capteurs

Ce sont des réseaux de capteurs qui sont déployés pour collecter des données sur l’environnement physique. Les capteurs peuvent mesurer des données telles que la température, l’humidité, la pression, la lumière, etc.

1.1.3 Les protocoles de communication

Ce sont des normes et des protocoles de communication qui permettent aux appareils connectés de communiquer entre eux et avec les systèmes de traitement de données.

1.1.4 Les plateformes IoT

Ce sont des plates-formes qui permettent de connecter, de gérer et de traiter des données provenant d’appareils connectés. Les plateformes IoT peuvent inclure des outils d’analyse de données, des services de stockage de données, des interfaces utilisateur, etc.

1.1.5 L’analyse de données

L’analyse de données est un processus qui permet d’extraire des informations utiles à partir des données collectées par les appareils connectés. Les algorithmes d’analyse de données peuvent identifier des tendances, des modèles et des anomalies dans les données qui peuvent être utilisées pour prendre des décisions et optimiser les processus.

1.1.6 La sécurité IoT

La sécurité IoT est un ensemble de pratiques et de technologies qui permettent de garantir la confidentialité, l’intégrité et la disponibilité des données collectées par les appareils connectés. Les enjeux de sécurité IoT sont de plus en plus importants à mesure que le nombre d’appareils connectés augmente.

1.1.7 Big data

L’IoT génère une quantité massive de données, appelées Big Data, provenant de capteurs et d’appareils connectés. L’analyse de ces données massives est essentielle pour tirer parti des avantages de l’IoT, tels que l’amélioration de l’efficacité énergétique, la sécurité routière et la surveillance de la santé. Cependant, l’analyse de ces données massives pose également des défis importants en termes de stockage, de traitement et de protection de la vie privée des utilisateurs. Les entreprises doivent être en mesure de stocker et de traiter ces données de manière efficace et sécurisée, tout en respectant les lois et les réglementations en matière de protection de la vie privée des utilisateurs.

1.2 Définition de l'IoT

L'IoT est l'interconnexion entre l'Internet et des objets, des lieux et des environnements physiques. L'appellation désigne un nombre croissant d'objets connectés à Internet permettant ainsi une communication entre nos biens dits physiques et leurs existences numériques. Ces formes de connexions permettent de rassembler de nouvelles masses de données sur le réseau.[1]

L'Internet des objets (IoT) est un écosystème constitué d'appareils connectés à Internet qui ont la capacité de collecter et de transférer automatiquement des données sur un réseau, sans nécessiter d'intervention humaine.

L'IoT permet aux objets de interagir avec leur environnement, qu'il soit interne ou externe, ce qui influe sur les processus de prise de décision.

La communication entre les appareils se fait via le réseau IP (Internet Protocol) en utilisant différents types de communication adaptés aux besoins spécifiques.

L'IoT trouve des applications dans divers domaines, tels que les villes intelligentes, l'agriculture intelligente, et bien d'autres, pour répondre aux besoins quotidiens des individus.[2]

1.3 Fonctionnement de l'IoT

L'Internet des objets (IoT) est un système complexe qui implique plusieurs étapes clés pour son fonctionnement efficace. Ces étapes incluent la collecte de données, le traitement des données, la transmission des données, l'analyse des données et enfin, la prise de décision.

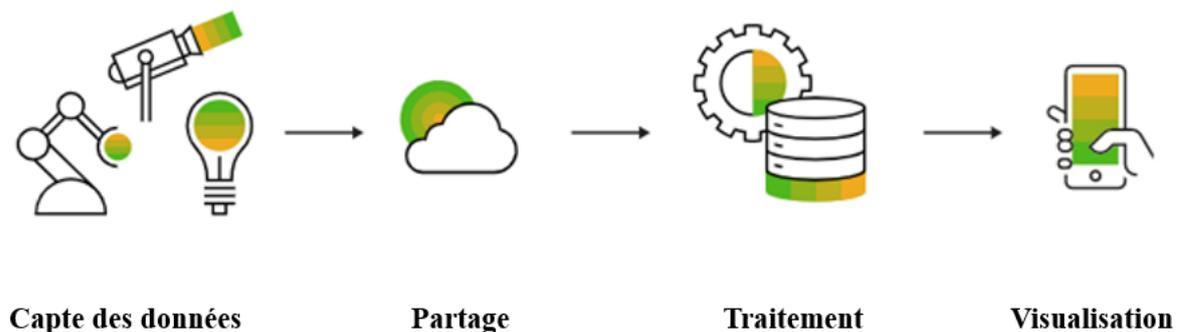


Figure 1.1 – Fonctionnement de l'IoT.

[3]

1.3.1 Collecte de données

Le dispositif IoT est équipé de capteurs pour collecter des données sur son environnement. Les capteurs peuvent mesurer des paramètres tels que la température, l'humidité, la pression, la position, la vitesse, etc.

1.3.2 Traitement de données

Les données collectées sont ensuite envoyées à un microcontrôleur ou à un processeur pour être traitées et analysées. Le traitement peut inclure la conversion de données analogiques en données numériques, le filtrage des données, la normalisation des données, la compression des données, etc.

1.3.3 Transmission de données

Les données traitées sont ensuite envoyées à un cloud ou à un système de stockage à distance via un réseau sans fil, tel que Wi-Fi, Bluetooth, Zigbee, etc.

1.3.4 Analyse de données

Les données collectées et stockées peuvent être analysées pour extraire des informations pertinentes et utiles. L'analyse peut inclure des techniques d'apprentissage automatique, de l'intelligence artificielle, de la visualisation de données, etc.

1.3.5 Prise de décision

Les informations obtenues grâce à l'analyse des données peuvent aider les utilisateurs ou les systèmes à prendre des décisions plus éclairées. Par exemple, un système de contrôle de la température peut ajuster automatiquement la climatisation en fonction des données de température collectées. [3]

1.4 Les fonctions de l'objet connecté

Un objet connecté est un dispositif qui possède des fonctionnalités mécaniques et/ou électriques propres, conçu soit dès le départ avec des capacités de connectivité, soit avec la possibilité d'ajouter une connectivité ultérieurement. [4]

L'objectif d'un objet connecté est de collecter des données à partir de capteurs, de traiter ces données, de communiquer par le biais de fonctionnalités de connectivité, et de recevoir des instructions pour exécuter des actions. Généralement, ces fonctionnalités de l'objet connecté requièrent de l'énergie, notamment lorsque les données sont prétraitées directement dans l'objet lui-même. [4]

1.4.1 Les capteurs

Un capteur est un appareil qui convertit une grandeur physique observée (température, lumière, mouvement, etc.) en une grandeur numérique exploitable par un logiciel.

Il existe une grande variété de capteurs de différents types, les objets connectés ont souvent la capacité de capter ces grandeurs physiques là où ils sont utilisés. Exemple

de capteurs : lumière, présence, proximité, position, déplacement, accélération, rotation, température, humidité, son, vibration, électrique, magnétique, chimique, gaz, flux, force, pression, niveau...[4]



Capteur de mouvement Capteur de température

Figure 1.2 – Capteur de mouvement et température
[5] [6]

1.4.2 Les sources d'énergies

Il existe 4 types de sources d'énergie : [4]

1 - Les sources d'énergie filaires

pour les objets ayant accès à une prise électrique.

2 - Les piles ou batteries

pour les objets inutilisables ou occasionnellement utilisés (recharge).



Figure 1.3 – Piles de type Duracell
[7]

3 - Les capteurs d'énergie

ou « energy harvesting » : (photovoltaïque, piézoélectrique, thermoélectrique, énergie cinétique, etc.) pour prolonger la durée de vie des objets à très faible consommation électrique.



Figure 1.4 – Capteur d'énergie
[8]

4 - Objets passifs alimentés

par les ondes électromagnétiques des lecteurs (RFID, NFC, etc.). L'énergie est l'un des grands enjeux des objets connectés, à la fois pour garantir la plus longue durée de vie possible sans maintenance et pour respecter l'environnement, malgré l'empiètement des objets connectés énergivores dans nos espaces personnels et professionnels. [4]

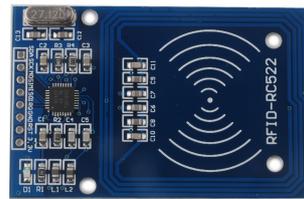


Figure 1.5 – Lecteur RFID
[9]

1.4.3 Les actionneurs

Les actionneurs sont des dispositifs qui convertissent des données numériques en phénomènes physiques pour produire du mouvement, ils sont un peu à l'opposé des capteurs. [4]
Exemple d'actionneurs : Afficheurs, Alarmes, Caméras, Haut-parleurs, Interrupteurs, Lampes, Moteurs, Pompes, Serrures, Vannes, Ventilateur, Vérins...



Figure 1.6 – Vérin pneumatique Alarme
[10]

1.4.4 La connectivité

La connectivité de l'objet est assurée par une petite antenne radiofréquence qui permet à l'objet de communiquer avec un ou plusieurs réseaux.

Ces objets sont capables de suivre des informations telles que leur identité, leur état, des alarmes ou des données de capteurs d'une part et de recevoir des informations telles que des commandes d'action et des données d'autre part.

Le module de connectivité peut également gérer le "cycle de vie des objets", c'est-à-dire l'authentification et l'enregistrement, le débogage, la mise à jour et la suppression dans le réseau.[4]

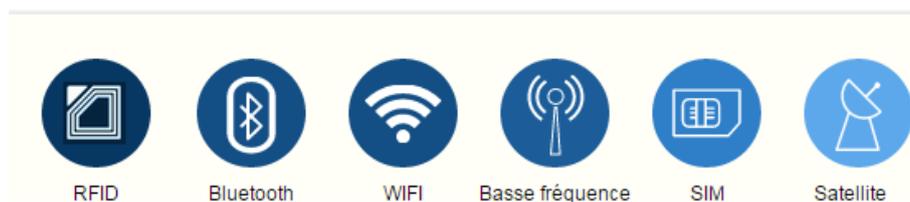


Figure 1.7 – Les technologies de connectivité.

[11]

1.5 Les protocoles de communication dans l'IoT

Le réseau IoT sert à doter un objet d'une connectivité à Internet pour permettre la remontée d'informations. Différents protocoles de communication sont disponibles sur le marché pour effectuer cela. Un protocole réseau est un langage de communication utilisé pour transmettre des informations. Il existe des protocoles de réseaux IoT, utilisés pour le développement des systèmes d'IoT, tels que :

1.5.1 Le Protocole HTTP

HTTP est un protocole largement utilisé sur Internet, notamment pour accéder aux sites Web. Il facilite l'échange de données entre les clients et les serveurs. Cependant, HTTP peut être considéré comme verbeux en raison de sa nature qui inclut une quantité importante d'informations et peut entraîner une consommation de données plus élevée. C'est là qu'intervient HTTPS, une version sécurisée de HTTP. En utilisant le chiffrement, HTTPS protège les éléments échangés entre les clients et les serveurs. Ainsi, les communications sont sécurisées et les données sont cryptées, offrant une meilleure confidentialité lors des échanges sur Internet. [12]

Dans le contexte de la communication client-serveur avec HTTP, le client envoie des requêtes HTTP au serveur, et le serveur répond en renvoyant des réponses HTTP au client.

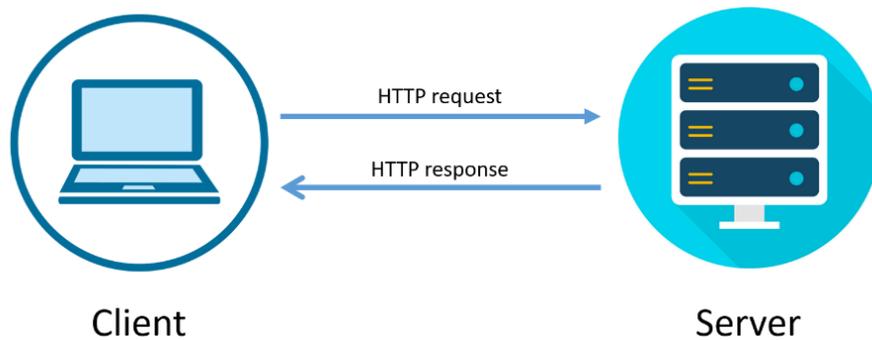


Figure 1.8 – Le protocole Http
[13]

1.5.2 Le Protocole MQTT

Queue Telemetry Transport MQTT c'est un protocole alternatif largement utilisé dans l'IoT, il est un protocole standardisé reposant sur TCP/IP. Il est particulièrement utilisé pour transporter des données des objets connectés sur le Cloud. Il est adapté aux connexions qui intègrent la partie mobilité entre le client et le serveur où seront stockées les données. Il fonctionne grâce à des "brokers"¹ (ou intermédiaires) qui centralisent les connexions entre différentes entités et leur permettent de s'abonner et/ou de publier des informations via des "topic"².

Ce mode de fonctionnement permet aux clients avec des connexions potentiellement changeantes ou des adresses IP variables, tels que des capteurs connectés via une connexion mobile. Il en va de même pour les serveurs. [12]

N.B : Dans le troisième chapitre, nous examinerons de manière approfondie le protocole MQTT.

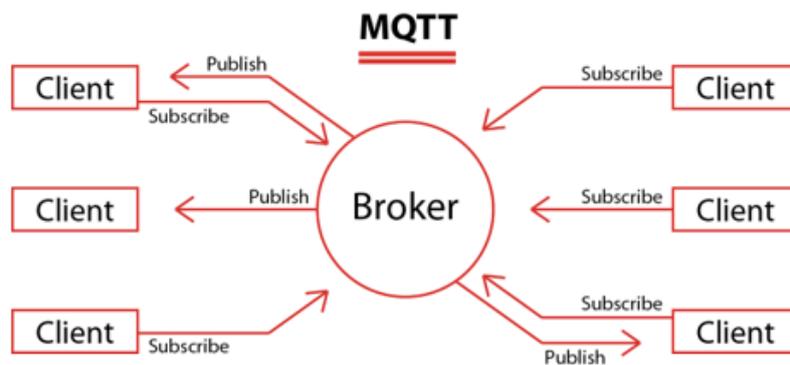


Figure 1.9 – MQTT Protocol
[14]

1. **BROKER** :un serveur intermédiaire qui facilite l'échange de messages entre les différentes parties. Il s'agit d'un composant central qui reçoit les messages des producteurs (ou émetteurs) et les distribue aux consommateurs (ou récepteurs) appropriés.

2. **TOPIC** : un canal de communication virtuel ou à un identifiant logique utilisé pour catégoriser et organiser les messages échangés entre les producteurs et les consommateurs.

1.5.3 Le Protocole CoAP

CoAP (Constrained Application Protocol) est un protocole destiné aux environnements contraints qui n'ont pas beaucoup de bande passante pour échanger des données. En fait, le nombre d'octets envoyés à chaque fois est limité et la surcharge du protocole est très faible. Ce protocole est très utile pour les environnements contraints avec des appareils à faibles capacités de traitement et de transmission. [12]

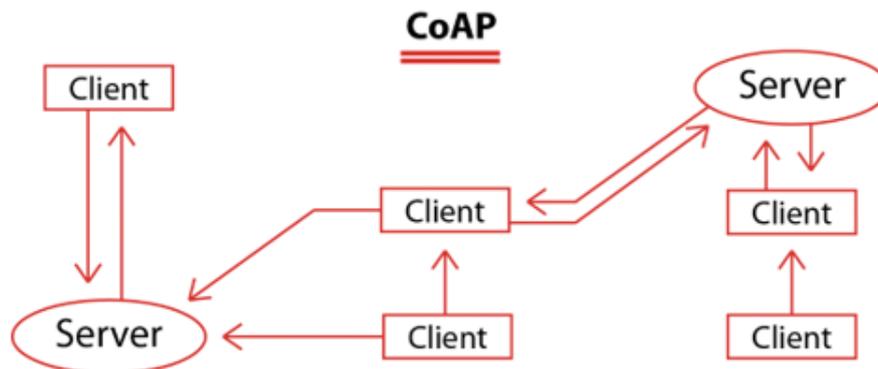


Figure 1.10 – CoAP Protocol
[15]

1.5.4 Comparaison entre HTTP, MQTT et CoAP

Pour obtenir une meilleure compréhension entre les protocoles, il est important de prendre en compte les aspects suivants :

1 -Architecture : Analyse de l'architecture client-serveur et comprendre comment les requêtes et les réponses sont échangées entre le client et le serveur.

2 -Légèreté : Comparaison de la taille des messages échangés par les protocoles, en tenant compte des en-têtes et de la quantité de données supplémentaires incluses.

3 -Fiabilité : Évaluation des mécanismes de gestion des erreurs, de détection et de récupération des erreurs de transmission, ainsi que les mécanismes de reprise après panne.

4 -Sécurité : Comparaison des mécanismes de sécurité offerts, tels que le chiffrement des données, l'authentification et la protection contre les attaques.

5 -Interopérabilité : Vérification de la capacité des protocoles à fonctionner de manière transparente entre différents systèmes et applications.

Voici un tableau qui présente ces divers aspects de comparaison.

Critère	HTTP	MQTT	CoAP
Architecture	Protocole de communication basé sur le modèle client-serveur	Protocole de messagerie de type publish-subscribe	Protocole de transfert de données entre les clients et les serveurs
Légèreté	Relativement lourd en raison de la syntaxe et des en-têtes supplémentaires	Léger en raison de la simplicité de la structure de message	Extrêmement léger grâce à l'utilisation de l'UDP et de structures de message minimalistes
Fiabilité	Fournit un mécanisme intégré de retransmission des messages et de vérification d'intégrité	Moins fiable que HTTP, mais offre des garanties de livraison grâce au mécanisme de retransmission de MQTT	Moins fiable que HTTP et MQTT, mais offre des mécanismes de retransmission et de confirmation
Sécurité	Prise en charge de HTTPS pour une communication sécurisée	Prise en charge de TLS pour une communication sécurisée	Prise en charge de DTLS pour une communication sécurisée
Interopérabilité	Très répandu et bien pris en charge par de nombreux frameworks et outils	Bien pris en charge par divers frameworks et bibliothèques, mais moins répandu que HTTP	Bien pris en charge par les frameworks et les outils spécifiques à CoAP, mais moins répandu que HTTP et MQTT

Tableau 1.1 – Comparaison entre les protocoles

[16]

1.5.5 Les technologies de communication dans l'IoT

Les moyens de communication dans l'IoT peuvent être classés en fonction de leur portée. Voici les trois principales catégories :

1 - Communication à courte portée

- Bluetooth : Utilisé pour la communication sans fil entre des appareils IoT à proximité les uns des autres.[12]
- Zigbee : Un protocole de communication basse consommation d'énergie et à faible portée, idéal pour les réseaux de capteurs.[17]
- NFC (Near Field Communication) : Utilisé pour des interactions de communication très courtes distances, souvent à moins de quelques centimètres.

2 - Communication à moyenne portée

- Wi-Fi : Utilisé pour les communications sans fil à moyenne portée, adapté aux

réseaux domestiques et aux applications industrielles.[12]

- Z-Wave : Un protocole de communication conçu pour les applications domotiques, offrant une portée moyenne et une faible consommation d'énergie.

3 - Communication à longue portée

- Cellular (3G, 4G, 5G) : Utilisé pour la communication à longue distance dans les applications IoT nécessitant une large couverture géographique.

- LoRaWAN : Un protocole de communication basse consommation d'énergie pour les réseaux à longue portée, adapté aux applications IoT étendues telles que les villes intelligentes et l'agriculture intelligente.

1.6 Architecture de l'IoT

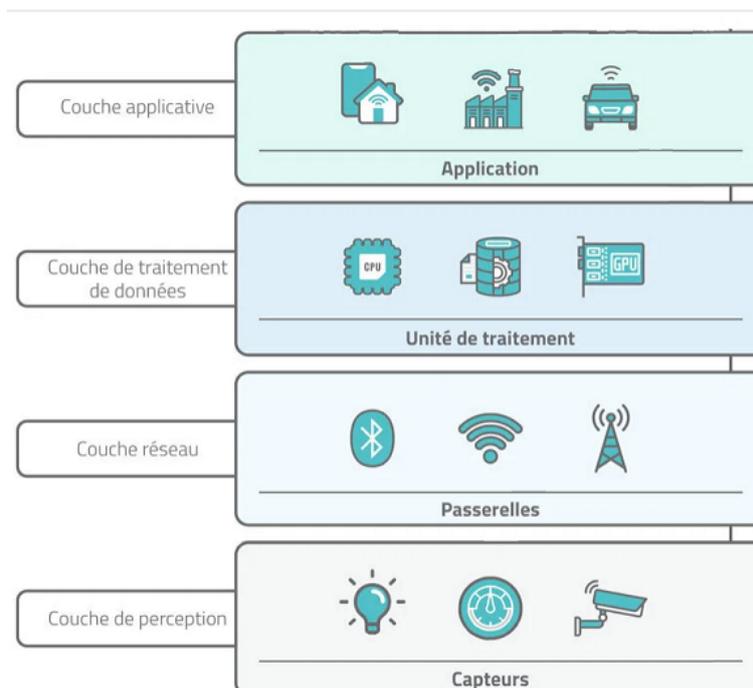


Figure 1.11 – Architecture de l'IoT.
[18]

L'architecture de l'IoT est une structure complexe qui permet la communication entre les différents composants de l'IoT. Elle se compose de plusieurs couches interconnectées qui permettent aux appareils de collecter des données, de les traiter et de les transmettre à des applications ou à des services cloud pour analyse et prise de décision. Voici les principales couches de l'architecture de l'IoT.

1.6.1 Couche de perception

Cette couche convertit les signaux analogiques en données numériques. Elle couvre un large éventail d'objets qui font le pont entre le monde réel et le monde numérique. Celles-ci se répartissent en 3 catégories :

- Capteurs tels que sondes, compteurs ou compteurs.
- Actionneurs pour commande de moteurs, lasers et bras robotisés.
- Machines et équipements connectés à des capteurs et actionneurs.

1.6.2 Couche réseau

La fonction principale de la couche réseau est de connecter l'appareil à d'autres objets intelligents, serveurs ou appareils pour traiter efficacement les données collectées. Cette couche gère également le transfert de toutes les données. Au niveau de la couche réseau, non seulement les passerelles internet et réseau sont responsables de la connexion entre le réseau de capteurs et internet. Il existe également des systèmes d'acquisition de données (DAS) qui fournissent des fonctions d'agrégation et de transformation des données. Parmi les technologies utilisées pour la transmission de données entre les objets connectés, on distingue : Wi-Fi, Bluetooth, Zigbee, NFC...

1.6.3 Couche de traitement des données

La couche de traitement a 3 tâches principales :

- Collecte des données en temps réel à partir de la couche réseau, permettant aux administrateurs de définir la pertinence et l'emplacement des données.
- Stocker ses données pertinentes et vraiment utiles dans une large gamme de solutions de stockage.
- Traitement des données pour améliorer l'interopérabilité des appareils intelligents.

1.6.4 Couche d'application

La couche applicative qui va interagir avec l'utilisateur via des services spécifiques.

Les applications peuvent être créées directement sur des plates-formes IoT qui fournissent une infrastructure de développement logiciel et des outils clés en main permettant l'exploration de données, l'analyse avancée et la visualisation de données. [19]

1.7 Les avantages de l'IoT

L'IoT, ou l'Internet des objets, est une technologie en plein essor qui permet la connectivité des objets physiques à Internet, offrant ainsi de multiples avantages dans divers domaines. Voici un aperçu des principaux avantages offerts par l'IoT :

L'IoT, ou l'Internet des objets, offre de nombreux avantages dans différents secteurs. Voici une explication détaillée de certains de ces avantages clés :

- **Optimisation des processus** : L'IoT permet l'automatisation et l'optimisation des processus dans divers domaines tels que l'industrie, l'agriculture et la logistique. Grâce à la collecte de données en temps réel par des capteurs, les processus peuvent être surveillés, analysés et ajustés pour améliorer l'efficacité, réduire les coûts et minimiser les erreurs.

- **Amélioration de la productivité** : L'IoT permet d'accroître la productivité en connectant les machines, les équipements et les outils pour une meilleure coordination et une utilisation plus efficace des ressources.

Par exemple, dans un environnement de fabrication, les machines peuvent communiquer entre elles pour optimiser la production et réduire les temps d'arrêt.

- **Amélioration de la qualité de vie** : L'IoT peut contribuer à améliorer la qualité de vie des individus en offrant des solutions intelligentes dans les domaines de la santé, de la domotique et des villes intelligentes.

Par exemple, des dispositifs médicaux connectés peuvent surveiller en temps réel les signes vitaux des patients et envoyer des alertes en cas de besoin, tandis que les maisons intelligentes offrent des fonctionnalités telles que le contrôle à distance des appareils et l'optimisation de la consommation d'énergie.

- **Prise de décisions éclairées** : L'IoT fournit des données en temps réel provenant de divers capteurs, ce qui permet une meilleure compréhension et une prise de décision éclairée. Ces données peuvent être analysées pour détecter des tendances, identifier des modèles et fournir des informations précieuses pour orienter les décisions stratégiques.

- **Réduction des coûts** : L'IoT peut contribuer à la réduction des coûts en optimisant l'utilisation des ressources et en évitant les gaspillages. Par exemple, dans l'agriculture, des capteurs intelligents peuvent surveiller l'humidité du sol et l'irrigation peut être ajustée en conséquence, ce qui réduit la consommation d'eau et les coûts associés.

- **Sécurité renforcée** : L'IoT offre des possibilités de renforcer la sécurité dans différents domaines. Par exemple, dans les systèmes de sécurité domestique, les caméras et les capteurs peuvent être connectés à des applications mobiles pour une surveillance en temps réel, et des alertes peuvent être envoyées en cas d'intrusion ou de situation anormale.

- **Innovation et développement de nouvelles applications** : L'IoT favorise l'innovation en ouvrant de nouvelles possibilités de développement d'applications et de services. Les développeurs peuvent exploiter les données collectées par les objets connectés pour créer des applications intelligentes et personnalisées répondant aux besoins spécifiques des utilisateurs.

1.8 Les défis de l'IoT

L'Internet des objets (IoT) présente également des défis importants qui doivent être pris en compte pour assurer son déploiement efficace et sécurisé. Voici quelques-uns des

principaux défis de l'IoT :

- **Sécurité des données** : Avec l'avènement des appareils IoT, les méthodes traditionnelles de sécurité des réseaux reposent sur : Les règles de pare-feu, les ACL, les VLAN et les VPN deviennent de plus en plus complexes, fastidieuses et cher à la minute.

Le changement de paradigme induit par l'IoT et son interaction avec les applications métiers nécessitent la sécurité des réseaux d'un point de vue de segmentation.

À mesure que l'Internet des Objets (IoT) deviendra une partie intégrante des processus métiers, de plus en plus d'applications et de données liées interagissent avec les dispositifs IoT. Ainsi, il est essentiel d'améliorer les mesures de sécurité et de créer des zones séparées dans le réseau pour empêcher toute violation de sécurité d'affecter les processus métiers ou les données. [13]

- **Intégration avec les systèmes existants** : La difficulté à connecter les objets connectés avec les systèmes existants est souvent due à deux causes principales. Tout d'abord, il y a la question des différents protocoles de communication utilisés.

Les objets connectés peuvent fonctionner avec des protocoles spécifiques tels que MQTT, CoAP, etc., qui ne sont pas directement compatibles avec les protocoles utilisés dans les systèmes déjà en place. Cela peut entraîner la nécessité de mettre en place des passerelles ou des adaptateurs pour permettre la communication entre les deux. Deuxièmement, les systèmes existants peuvent être complexes, comprenant plusieurs couches d'infrastructure et de sécurité.

L'intégration des objets connectés dans ces systèmes peut exiger des modifications importantes afin d'assurer leur compatibilité. Il est donc essentiel de prendre en compte ces facteurs lors de la mise en place de l'IoT afin de garantir une intégration fluide et efficace.[16]

- **Le coût élevé** : L'Internet des objets (IoT) est une technologie émergente qui permet la connectivité des objets physiques tels que les appareils électroménagers, les voitures, les capteurs, etc., à Internet.

Bien que l'IoT présente de nombreux avantages en termes d'efficacité, de productivité et de commodité, il peut également entraîner des coûts élevés. Ces coûts peuvent être attribués à plusieurs facteurs.

Tout d'abord, les coûts matériels liés aux capteurs, aux appareils et aux modules de communication nécessaires pour les solutions IoT peuvent être élevés.

Deuxièmement, les coûts de connectivité peuvent être considérables en raison des frais de transfert de données et de l'utilisation d'un réseau dédié.

Troisièmement, la mise en place de mesures de sécurité pour les solutions IoT est essentielle, mais cela peut engendrer des coûts importants en termes de ressources et de temps.

L'IoT présente également plusieurs défis tels que la sécurité, l'interopérabilité, la gestion de l'énergie, la gestion des données, les coûts et la protection de la vie privée.

Cependant, ces défis peuvent être surmontés grâce à une conception appropriée, à une utilisation responsable et à une collaboration entre les différentes parties prenantes. Il est essentiel de trouver un équilibre entre les avantages de l'IoT et les coûts associés, tout en

veillant à la sécurité, à la protection de la vie privée et à la durabilité de cette technologie en constante évolution.[16]

1.9 Les enjeux éthiques et environnementaux de l'IoT

L'Internet des Objets (IoT) a le potentiel de transformer nos vies en permettant une connectivité accrue entre les appareils physiques et Internet. Cependant, cette technologie soulève également des préoccupations éthiques et environnementales importantes qui doivent être prises en compte.

- **Impact environnemental** : En IoT, le facteur d'impact le plus visible est son ampleur. Selon des estimations, en fin d'année 2021, il y avait entre 31 et 36 milliards de dispositifs IoT en circulation, mais les prévisions suggèrent que ce nombre pourrait atteindre 75 milliards d'ici 2025 et même 125 milliards d'ici la fin de la décennie.

Cela pourrait causer une augmentation significative de la quantité de déchets électroniques, ce qui pourrait présenter un défi environnemental considérable. Les dispositifs intelligents posent deux problèmes spécifiques qui aggravent le problème déjà considérable des déchets électroniques.

Tout d'abord, les fabricants ajoutent des semi-conducteurs à des produits qui n'en avaient pas besoin auparavant, ce qui réduit leur durée de vie. Par exemple, des produits qui avaient une durée de vie de 15 ans sont maintenant remplacés au bout de moins de cinq ans. En outre, de nombreux dispositifs connectés de petite taille tels que les trackers, les bijoux et les dispositifs portables sont souvent considérés comme étant hors d'usage lorsque leur batterie se décharge. Dans ces cas-là, les consommateurs les jettent et en achètent de nouveaux, ce qui contribue également à augmenter la quantité de déchets électroniques. [20]

- **Protection de la vie privée** : La sécurité des données est un élément clé pour établir la confiance des utilisateurs, qu'ils soient particuliers ou professionnels, vis-à-vis des objets connectés. Bien que les utilisateurs puissent être prudents quant à l'utilisation de leurs données par les opérateurs qui les collectent, leur méfiance est encore plus grande en ce qui concerne les tiers et surtout les risques de piratage. La sécurité doit donc être envisagée dès la conception de l'objet connecté lui-même.[21]
- **Équité et justice sociale** : Le risque de creuser le fossé entre les communautés qui ont accès à l'IoT et celles qui n'en ont pas est lié aux inégalités économiques et technologiques. En effet, les objets connectés sont souvent associés à des coûts élevés, tant pour leur acquisition que pour leur utilisation, ce qui peut rendre leur accès difficile pour certaines personnes ou communautés moins aisées sur le plan économique. Si une partie de la population a accès à l'IoT et bénéficie de ses avantages, comme une meilleure qualité de vie, une meilleure santé ou une productivité accrue, alors que l'autre partie n'en bénéficie pas, cela peut entraîner une augmentation des inégalités économiques et sociales. Cela peut également créer

une situation où une partie de la population dispose d'une quantité de données importante, tandis que l'autre n'a pas accès à ces données, ce qui peut aggraver les inégalités en matière d'information et de prise de décision.[16]

1.10 Les perspectives d'avenir de l'IoT

L'IoT est une technologie qui continue de se développer et d'évoluer à un rythme rapide. Ses perspectives d'avenir sont donc très prometteuses, on distingue :

1.10.1 Évolution des technologies IoT (Intelligence Artificielle IoT)

L'IAoT (ou AIIoT) est une combinaison de l'IA et de l'IoT qui permet d'améliorer le fonctionnement des objets connectés, d'améliorer les interactions entre les humains et les machines, et d'approfondir la gestion des données et l'analyse. Les capacités d'apprentissage automatique de l'IA sont combinées avec la connectivité, la signalisation et l'échange de données de l'IoT pour enrichir et transformer les deux technologies. Avec le développement progressif des réseaux IoT dans les grands secteurs de l'économie, la quantité de données non structurées disponibles augmentera automatiquement, permettant des solutions d'analyse de données plus avancées. L'union de l'IA et de l'IoT peut servir de base à ces solutions qui valorisent les données des objets connectés. Par exemple :

- **Les véhicules autonomes :**

L'IAoT peut jouer un rôle crucial dans les véhicules autonomes en combinant l'intelligence artificielle et l'Internet des objets pour rendre les voitures plus intelligentes et plus autonomes. Les voitures autonomes sont équipées de nombreux capteurs et systèmes qui collectent des données en temps réel sur leur environnement, tels que la position, la vitesse, la direction, la distance entre les voitures, les conditions météorologiques, les panneaux de signalisation, etc. L'IAoT utilise ces données pour analyser et comprendre l'environnement et prendre des décisions en conséquence, telles que la décision de freiner, de tourner ou d'accélérer.[22]

1.10.2 Développement de nouvelles applications

L'Internet des objets (IoT) ouvre de nouvelles perspectives pour l'industrie, en permettant la collecte et l'analyse de données en temps réel, la surveillance à distance et la prise de décisions plus éclairées. Voici quelques exemples de nouvelles applications industrielles développées grâce à l'IoT :

- **Maintenance prédictive :** les capteurs IoT intégrés aux machines peuvent fournir des informations sur les performances en temps réel, ce qui permet de prévoir les défaillances et les temps d'arrêt, d'anticiper les besoins de maintenance et de planifier les interventions avant que les pannes ne surviennent.

- **Optimisation de la chaîne d'approvisionnement :** l'IoT permet de suivre la chaîne d'approvisionnement à chaque étape, de la production au transport, en passant

par le stockage et la distribution. Cela permet de mieux gérer les niveaux de stock, d’optimiser la logistique et de réduire les coûts.[23]

1.10.3 Implications sociétales de l’IoT

L’IoT (Internet des objets) a des implications sociétales importantes, car il modifie la façon dont nous interagissons avec notre environnement physique et numérique. Voici des implications sociétales clés de l’IoT :

- **Connectivité accrue** : L’IoT permet une connectivité accrue entre les appareils, les personnes et les systèmes, ce qui peut améliorer l’efficacité et la commodité de la vie quotidienne. Par exemple, les appareils domestiques intelligents peuvent être contrôlés à distance à l’aide d’un smartphone, ce qui permet de gérer facilement les tâches ménagères.
- **Vie privée** : La collecte et l’utilisation de données par les appareils IoT soulève des préoccupations en matière de vie privée. Les données peuvent être utilisées pour suivre les comportements et les habitudes des utilisateurs, ce qui peut être utilisé à des fins publicitaires, mais également pour des pratiques plus malveillantes telles que le vol d’identité. [16]

En conclusion, les perspectives d’avenir de l’IoT sont très positives et nous pouvons nous attendre à ce que l’IoT continue de se développer et de s’améliorer dans les années à venir, en offrant des avantages considérables pour les entreprises et les consommateurs, tout en relevant les défis liés à la sécurité et à la protection de la vie privée.

1.11 L’état de l’art

L’Internet des objets (IoT) est le réseau d’objets physiques - dispositifs, instruments, véhicules, bâtiments et autres éléments intégrés avec de l’électronique, des circuits, des logiciels, des capteurs et une connectivité réseau - qui permet à ces objets de collecter et d’échanger des données. L’Internet des objets permet aux objets d’être détectés et contrôlés à distance via l’infrastructure réseau existante, offrant ainsi des opportunités d’intégration plus directe du monde physique dans les systèmes informatiques et permettant une amélioration de l’efficacité et de la précision. Le concept d’un réseau d’appareils intelligents a été discuté dès 1982, avec une machine à soda modifiée à l’Université Carnegie Mellon devenant le premier appareil connecté à Internet [24], capable de signaler son inventaire et si les boissons nouvellement chargées étaient froides. Kevin Ashton (né en 1968) est un pionnier britannique de la technologie connu pour avoir inventé le terme ”l’Internet des objets” pour décrire un système où l’Internet est connecté au monde physique via des capteurs omniprésents.

L’IoT est capable d’interagir sans intervention humaine. Certaines applications préliminaires de l’IoT ont déjà été développées dans les secteurs de la santé, des transports et de l’automobile. Les technologies de l’IoT en sont encore à leurs premiers stades de développement ; cependant, de nombreuses nouvelles avancées ont eu lieu dans l’intégration d’objets avec des capteurs sur Internet. Le développement de l’IoT soulève de nombreuses questions telles que l’infrastructure, les communications, les interfaces, les protocoles et les normes.

de plus, plusieurs études sont réalisées dans ce contexte, prenant l'exemple d'Une étude réalisée en 2014 a révélé que près de 20 % des entreprises mondiales interrogées (soit 1500 entreprises) investissent désormais dans l'Internet des objets (IoT), comparé à seulement 17 % en 2013. De plus, 14 % de ces entreprises ont indiqué que la priorité pour les cinq prochaines années était de mettre en place des capteurs sensoriels. Un exemple concret de cette tendance est l'opérateur ATT, qui a développé un système de "bagages intelligents" utilisant des technologies LTE et GPS. Ainsi, il est maintenant possible de suivre ses bagages en temps réel à l'aide d'un smartphone ou d'une tablette. De même, la ville de San Jose a mis en place un réseau de capteurs pour mesurer le trafic, ainsi que la pollution atmosphérique et sonore.[25] Prenons l'exemple de Washington, DC, où la ville a collaboré avec des développeurs externes au gouvernement pour créer une application permettant aux citoyens de signaler en temps réel les nids-de-poule à l'aide de leur téléphone intelligent. Cette initiative visait à améliorer la capacité de la ville à détecter et résoudre les problèmes sur ses routes.

Grâce à cette application, la ville a pu réaliser des économies significatives dès la première année. Les économies totales s'élèvent à 2,5 millions de dollars, représentant les coûts évités liés à une intervention plus rapide et plus ciblée pour réparer les nids-de-poule signalés par les citoyens. Cette approche collaborative entre les citoyens et le gouvernement local a permis d'optimiser les ressources et de garantir des routes plus sûres et plus agréables pour tous. [26] Ainsi, cette nouvelle réalité entraîne une phase de changement importante. La gestion efficace du vaste flux de services sur l'Internet nécessite la création et la disponibilité de nouveaux outils de découverte plus performants et précis [27]. Dans ce contexte, il est essentiel de souligner l'importance de la phase de découverte dans le domaine de l'Internet des objets (IoT). Cela constitue l'un des premiers principes de la raison pratique sur lesquels il convient de porter une attention particulière. Des chercheurs ont également souligné l'importance d'orienter l'évolution du Web pour servir les intérêts de l'IoT [28] [29]. Ils suggèrent de commencer par Hadjila et de poursuivre dans cette voie.

Dans le domaine de la gestion et du contrôle d'accès aux laboratoires, plusieurs modalités d'identification sont utilisées pour assurer la sécurité et la gestion efficace de l'accès. Une recherche menée par Oussama Afouf [30] en 2020 a exploré ces modalités dans le contexte d'un Smart Laboratory au sein d'une Smart University. Les modalités d'identification incluaient les codes QR, les RFID, la reconnaissance faciale, le clavier et l'empreinte digitale. Ces méthodes permettent de vérifier l'identité des personnes autorisées à accéder au laboratoire, renforçant ainsi la sécurité et facilitant la gestion du contrôle d'accès.

L'étude réalisée par Jean Paul EZIKOLA MAZOBA[31] en 2016 porte sur l'Internet des objets (IoT) et le contrôle d'accès aux données. L'objectif principal de cette étude était de fournir une description approfondie des aspects généraux du réseau Internet et de ses fonctionnalités, ainsi que des aspects techniques de l'Internet des objets et du contrôle du flux de données, y compris les moyens d'accès aux données. Dans cette étude, une attention particulière a été accordée à la compréhension des fonctionnalités et des caractéristiques techniques de l'Internet des objets.

En ce qui concerne le contrôle d'accès aux données, l'étude s'est concentrée sur les mécanismes permettant de réguler l'accès aux informations générées par les dispositifs de

l'Internet des objets. Cela comprend la gestion des autorisations, la sécurité des données et les méthodes de contrôle du flux de données.

L'étude visait à fournir une vue d'ensemble des connaissances actuelles dans le domaine de l'Internet des objets et du contrôle d'accès aux données. Elle a exploré les différentes fonctionnalités et possibilités offertes par le réseau Internet, ainsi que les techniques avancées spécifiques à l'Internet des objets. L'objectif était d'informer sur les concepts clés, les défis et les opportunités associés à l'exploitation de l'Internet des objets et à la gestion du flux de données.

Dans le domaine des appareils IoT à base de composants en réseau, Ngoc Hai BUI [32] a réalisé un état de l'art axé sur un mécanisme de mise à jour de logiciels écoénergétiques. L'objectif principal de cette étude était de proposer une approche qui permet de planifier les mises à jour de tous les appareils d'un réseau IoT afin de minimiser leur consommation d'énergie.

L'étude a mis l'accent sur la distribution efficace des composants logiciels au sein du réseau, en tenant compte des contraintes énergétiques des appareils IoT. Différentes stratégies de distribution ont été analysées, notamment en termes de priorisation des mises à jour en fonction de leur impact sur la consommation d'énergie.

Rabeb Saad[33] qui a travaillé sur le sujet intitulé ; Modèle collaboratif pour l'Internet des objets (IoT) en 2016. Son travail a porté sur un modèle théorique qui propose un ensemble de primitives et une nouvelle stratégie de 29 collaborations afin de partager les données dans le monde de l'internet des objets. L'étude s'est concentrée sur la stratégie de décentralisation, mettant en avant les avantages et les défis liés à ce type de modèle collaboratif. L'accent a été mis sur la conception d'un langage spécifique appelé IoTCollab, qui vise à faciliter la programmation et l'intégration des différents concepts introduits par les modèles de partage de données.

Selon HANAFI Imane et YAHIA Samia[34] qui ont fait une Conception d'un système IoT de bout en bout pour le suivi temps réel des processus de production, de géolocalisation et de surveillance en 2020. L'objectif principal de cette étude était de développer une application web permettant le suivi en temps réel des grandeurs environnementales, telles que la température et l'humidité. Cette application a été conçue pour aider les utilisateurs à visualiser les données acquises par le capteur DHT-22 sous forme de graphes en temps réel. Cette étude a mis l'accent sur la réalisation d'un système complet basé sur l'Internet des objets (IoT). Il a examiné les différentes étapes de conception, de la collecte des données à leur visualisation en temps réel. L'étude a également porté sur la géolocalisation et la surveillance des processus de production, en mettant en évidence l'importance de ces fonctionnalités dans un contexte industriel.

Conclusion

En conclusion, le premier chapitre consacré aux généralités de l'IoT a été essentiel pour acquérir une compréhension approfondie de cette technologie en plein essor. Nous avons exploré les concepts fondamentaux de l'IoT, mis en évidence ses avantages et ses applications dans divers domaines, et examiné les défis et les limites qui lui sont associés,

notamment en termes de sécurité et de vie privée.

Le deuxième chapitre se concentrera sur la gestion des données IoT, avec une attention particulière portée aux bases de données chronologiques. Nous plongerons dans des sujets tels que les capteurs, les protocoles de communication et les plateformes de gestion de données, afin de comprendre en profondeur la collecte, le stockage et le traitement des données dans un contexte chronologique. De plus, nous explorerons les méthodes d’analyse et de traitement des données de séries chronologiques, offrant ainsi des perspectives précieuses pour extraire des informations exploitables à partir de l’énorme quantité de données générées par les dispositifs IoT.

Ce chapitre sera d’une grande utilité pour les professionnels travaillant dans le domaine de l’IoT, leur permettant d’approfondir leurs connaissances et de développer une expertise pratique en matière de gestion des données IoT. Les informations et les concepts abordés contribueront à renforcer leur capacité à relever les défis liés à la gestion et à l’exploitation efficaces des données IoT, en vue de maximiser les avantages potentiels offerts par cette technologie.

En somme, ce deuxième chapitre jouera un rôle clé dans la consolidation de notre compréhension de l’IoT et de son fonctionnement, et fournira une base solide pour les chapitres à venir, qui exploreront d’autres aspects cruciaux de cette passionnante révolution technologique.

Chapter 2

Les séries chronologiques

Introduction

La gestion des données IoT constitue une exploration approfondie, en mettant l'accent sur les bases de données de séries chronologiques. Les séries chronologiques jouent un rôle crucial dans la collecte, le stockage et l'analyse des données générées par les dispositifs IoT, offrant ainsi des perspectives précieuses pour comprendre les tendances, les modèles et les prévisions.

Avant de plonger dans le domaine des séries chronologiques, il est important de comprendre les différents types de données qui sont collectées et traitées dans le contexte de l'IoT. Les données générées par les dispositifs IoT peuvent être de différentes natures, allant des données numériques classiques telles que les mesures de température, d'humidité ou de pression, aux données plus complexes comme les images, les vidéos ou les données géospatiales.

L'analyse et la gestion de ces différents types de données nécessitent des approches spécifiques adaptées à leurs caractéristiques uniques. Par exemple, les données chronologiques nécessitent une gestion particulière en raison de leur nature temporelle, tandis que les données géospatiales nécessitent des techniques de traitement spécifiques pour exploiter leurs informations de localisation.

De plus, l'intelligence artificielle (IA) joue un rôle de plus en plus important dans l'exploitation des données IoT. L'utilisation de techniques d'apprentissage automatique et de modèles prédictifs permet aux systèmes IoT d'identifier des modèles cachés, de prendre des décisions intelligentes et de fournir des services avancés. L'IA est étroitement liée à la gestion des données de séries chronologiques, car elle permet d'extraire des informations significatives à partir de ces données et d'améliorer les capacités de prédiction et d'optimisation.

Dans ce chapitre, nous explorerons donc les différentes dimensions des données IoT, en mettant l'accent sur les séries chronologiques. Nous aborderons les concepts fondamentaux des bases de données de séries chronologiques, leur utilisation et leurs applications dans le domaine de l'IoT. De plus, nous explorerons les liens entre les données de séries chronologiques et l'intelligence artificielle, en mettant en évidence les possibilités offertes par ces synergies.

En somme, ce chapitre fournira aux lecteurs une compréhension approfondie des données IoT et de leur gestion dans le contexte des séries chronologiques. Il jettera les bases nécessaires pour aborder les méthodes d'analyse et de traitement des données de séries chronologiques dans les chapitres suivants, en vue d'exploiter pleinement le potentiel des données IoT et de l'IA pour des applications diverses et novatrices.

2.1 Définition d'une donnée

La définition d'une donnée peut varier en fonction des perspectives et des domaines de recherche. Voici quelques définitions proposées par des chercheurs renommés dans le domaine: Selon William S. Cleveland [35] "Une donnée est une valeur ou un ensemble de

valeurs obtenues par mesure, observation ou expérimentation.”

Selon Peter Naur [36] ”Une donnée est un ensemble de symboles qui représentent des faits, des concepts ou des instructions, appropriés pour la communication, l’interprétation ou le traitement par des humains ou des machines.”

Selon David L. Donoho [37] ”Une donnée est une représentation codée d’une information ou d’une idée, pouvant être stockée, traitée et transmise.”

Et selon C.F. Jeff Wu [38] ”Une donnée est une information brute qui est collectée ou générée dans le but d’être utilisée pour prendre des décisions, résoudre des problèmes ou découvrir des connaissances.”

Ces définitions soulignent l’importance des données en tant que représentation de faits ou d’informations qui peuvent être utilisées dans différents contextes pour prendre des décisions, résoudre des problèmes ou générer des connaissances.

2.2 Les types de données

On a plusieurs types de données, chacun présentant des caractéristiques distinctes et des méthodes d’analyse spécifiques. Parmi ces types de données, nous pouvons mentionner:

2.2.1 Les données transversales

Également connues sous le nom de données en coupe transversale, représentent une observation ponctuelle à un moment précis dans le temps. Elles sont collectées en rassemblant des données de différents individus, objets ou unités à un instant donné.

Exemple

- Une enquête démographique qui collecte des informations telles que l’âge, le sexe, le niveau d’éducation et le revenu de différents individus à un moment précis.
- Un recensement de la population qui recueille des données sur la taille de la population, la répartition géographique et les caractéristiques socio-économiques des résidents à un instant donné.

2.2.2 Données longitudinales (Longitudinal data)

Les données longitudinales sont collectées sur une période de temps spécifique et permettent de suivre les mêmes individus, objets ou unités à différentes étapes. Elles fournissent des informations sur l’évolution ou les changements au fil du temps.

Exemple

- Une étude médicale qui suit un groupe de patients sur plusieurs années pour évaluer l’efficacité d’un traitement et observer les changements dans leur état de santé.
- Une étude longitudinale sur l’éducation qui suit un groupe d’élèves depuis leur entrée à l’école jusqu’à leur graduation, en collectant des données sur leurs performances scolaires et leurs résultats d’apprentissage tout au long de cette période.

2.2.3 Données catégoriques (Categorical data)

Les données catégoriques regroupent les observations en catégories ou en classes distinctes. Elles sont utilisées pour représenter des variables qualitatives ou des attributs non numériques.

Exemple

- La couleur des yeux d'un groupe de personnes, qui peut être classée en catégories telles que bleu, vert, marron, etc.
- Les résultats d'un sondage politique où les participants sont classés en catégories telles que "en faveur", "opposés" ou "sans opinion".

2.2.4 Données numériques (Numerical data)

Les données numériques sont des valeurs quantitatives mesurables qui peuvent être exprimées sous forme de nombres. Elles permettent des calculs mathématiques et des analyses statistiques.

Exemple

- La taille en centimètres d'un échantillon d'individus.
- Le revenu annuel en dollars d'un groupe de ménages.
- Le nombre de ventes réalisées par un commerçant chaque mois.

2.2.5 Données temporelles (Time series data)

Les données temporelles sont collectées à des intervalles réguliers au fil du temps, ce qui permet d'analyser les variations et les tendances sur une période donnée.

Exemple

- Les données météorologiques collectées chaque heure, telles que la température, l'humidité et la pression atmosphérique.
- Les données financières quotidiennes des cours des actions d'une entreprise.
- Les données de trafic routier collectées à intervalles réguliers pour analyser les modèles de congestion.[39]

Dans ce travail, notre intérêt se concentre principalement sur les données de séries chronologiques. Nous nous focalisons sur l'étude des méthodes et des techniques spécifiques associées à ce type de données.

2.3 Définition d'une série chronologique

Selon Morris Hamburg[40] "Une série chronologique est un ensemble d'observations statistiques classées par ordre chronologique. " Les données de séries chronologiques se concentrent sur les observations d'un seul individu à différents moments, généralement à des intervalles uniformes, la collecte de ces observations est appelée données de séries chronologiques.

Il s'agit des données de la même variable sur une période de temps telle que des mois, des trimestres, des années, etc. Les données de série chronologique se présentent sous la forme de X_t . Telle que t représente le temps.

Une série chronologique peut également être appelée série temporelle ou chronique.

2.4 Les exigences des séries chronologiques

Les exigences des séries chronologiques se réfèrent aux caractéristiques et aux conditions nécessaires pour traiter et analyser efficacement ce type de données. Voici quelques-unes des principales exigences des séries chronologiques :

2.4.1 Régularité temporelle

Les observations doivent être effectuées à des intervalles de temps réguliers pour que les données puissent être considérées comme une série chronologique. Cela permet d'observer les motifs temporels et les tendances

2.4.2 Ordre temporel

Les observations doivent être enregistrées dans l'ordre chronologique dans lequel elles ont été collectées. Cela garantit que l'analyse des données se fait dans le contexte temporel approprié.

2.4.3 Stationnarité

La stationnarité est une propriété des séries chronologiques dans laquelle les caractéristiques statistiques, telles que la moyenne et la variance, restent constantes dans le temps. Elle simplifie l'analyse et facilite la prévision.

2.4.4 Absence de valeurs manquantes

Les séries chronologiques doivent idéalement être exemptes de valeurs manquantes. Les données manquantes peuvent avoir un impact sur l'analyse et nécessitent des techniques spécifiques pour les combler ou les gérer.

2.4.5 Homoscédasticité

L'homoscédasticité se réfère à l'homogénéité de la variance des résidus dans une série chronologique. Cela signifie que la dispersion des valeurs résiduelles reste constante dans le temps.

2.4.6 Absence de tendances saisonnières

Dans certains cas, les séries chronologiques peuvent présenter des tendances saisonnières, c'est-à-dire des variations régulières qui se répètent à des intervalles fixes. La détection et la correction de ces tendances sont essentielles pour une analyse précise.

2.5 Représentation graphique des données d'une série temporelle

Les graphes de séries chroniques sont représentés par deux axes, un axe par rapport au temps et l'autre pour la variable, les points $(t ; Y_t)$ sont reliés par des segments de droites

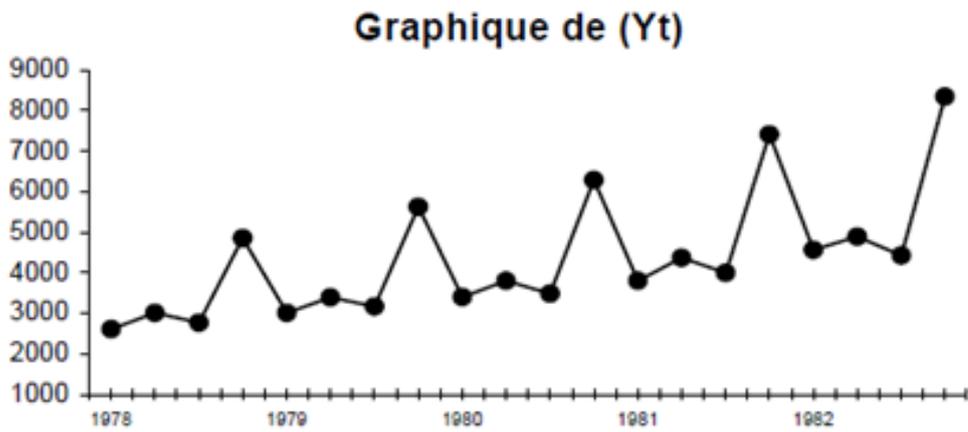


Figure 2.1 – Représentation graphique des données d'une série temporelle.
[41]

2.6 Base de données des séries chronologique

La Base de données des séries chronologiques constitue un outil essentiel pour la gestion et l'analyse de données temporelles, offrant une structure organisée pour capturer et explorer l'évolution des informations au fil du temps.

2.6.1 Définition d'une base de données des séries chronologiques

Une base de données des séries chronologiques est un système qui permet de stocker, organiser et gérer des ensembles de données temporelles. Elle est spécifiquement conçue pour traiter et analyser les séries chronologiques, en offrant des fonctionnalités adaptées à leur nature temporelle.[42]

Les bases de données des séries chronologiques sont conçues pour gérer de grandes quantités de données temporelles, souvent collectées en temps réel ou à intervalles réguliers. Elles sont optimisées pour stocker efficacement les données temporelles et permettre des opérations rapides de recherche, d'insertion, de mise à jour et de suppression.

Ces bases de données offrent également des fonctionnalités avancées pour l'analyse des séries chronologiques, telles que des algorithmes de compression et de réduction de données, des outils de visualisation, des techniques de traitement et d'agrégation temporelle, ainsi que des capacités de requêtage spécifiques aux séries chronologiques.

Certaines bases de données des séries chronologiques sont conçues pour prendre en charge des fonctionnalités spécifiques, telles que la gestion de données IoT (Internet of Things) ou la prévision temporelle. Elles peuvent également offrir des intégrations avec d'autres outils et plateformes d'analyse, facilitant ainsi l'exploration et l'exploitation des données temporelles.

2.6.2 Importance du Time Series Database (TSDB)

Les bases de données des séries chronologiques (TSDB - Time Series Database) jouent un rôle crucial dans la gestion et l'analyse des données temporelles. Elles offrent plusieurs avantages et fonctionnalités qui les rendent indispensables dans de nombreux domaines. Voici quelques points soulignant l'importance des TSDB :

- **Stockage optimisé des données temporelles :** Les TSDB sont spécialement conçues pour stocker efficacement les données temporelles, ce qui permet de gérer de grandes quantités de données sans compromettre les performances. Elles utilisent des structures de données et des algorithmes adaptés aux séries chronologiques, permettant un stockage compact et une récupération rapide des données.
- **Analyse avancée des séries chronologiques :** Les TSDB proposent des fonctionnalités d'analyse avancée spécifiquement adaptées aux séries chronologiques. Elles offrent des outils de traitement, de visualisation et d'agrégation temporelle, permettant d'explorer les données, d'identifier des motifs, de détecter des anomalies et de réaliser des prévisions.
- **Gestion des données en temps réel :** Les TSDB sont souvent utilisées pour stocker des données en temps réel provenant de capteurs, de dispositifs IoT ou d'autres sources. Elles permettent une ingestion rapide des données et une mise à jour en temps réel, facilitant ainsi le suivi et la surveillance en temps réel des variables et des événements temporels.
- **Évolutivité et performances :** Les TSDB sont conçues pour gérer des volumes massifs de données temporelles et offrir des performances élevées, même avec des requêtes complexes sur de larges périodes. Elles peuvent être mises à l'échelle horizontalement pour s'adapter à des charges de travail croissantes et à des données en expansion, garantissant ainsi des performances optimales.
- **Intégration avec d'autres outils d'analyse :** Les TSDB peuvent être intégrées à d'autres outils et plateformes d'analyse de données, facilitant ainsi l'intégration des données temporelles dans des flux de travail plus vastes. Elles peuvent être connectées à des outils de visualisation, de modélisation prédictive ou d'apprentissage automatique,

permettant une analyse plus approfondie et des insights précieux.

2.6.3 Différence entre les BDD temporelles et les BDD relationnelles

Les bases de données temporelles (TSDB) et les bases de données relationnelles (RDB) présentent des différences majeures en termes de structure, de fonctionnalités et d'utilisation. Voici quelques différences clés entre les deux types de bases de données, accompagnées d'exemples :

1 - Structure des données

- **Bases de données temporelles :** Les TSDB sont spécifiquement conçues pour stocker et gérer des données temporelles, telles que des séries chronologiques. Elles utilisent une structure de données optimisée pour les opérations temporelles, avec une indexation temporelle et une gestion efficace des valeurs temporelles.

Exemple: InfluxDB est une base de données temporelle largement utilisée pour stocker des mesures de capteurs IoT.

- **Bases de données relationnelles :** Les RDB stockent des données structurées dans des tables avec des relations entre elles. Elles sont adaptées à une grande variété de données, y compris les données transactionnelles et les informations interconnectées.

Exemple: MySQL est une base de données relationnelle populaire utilisée dans de nombreux domaines, notamment les sites web et les applications d'entreprise.

2 - Fonctionnalités

- **Bases de données temporelles :** Les TSDB offrent des fonctionnalités avancées pour la gestion des données temporelles, telles que des agrégations temporelles, des fenêtres de temps, des requêtes temporelles et des prévisions.

Exemple: TimescaleDB est une base de données temporelle basée sur PostgreSQL, offrant des fonctionnalités étendues pour l'analyse des données temporelles.

- **Bases de données relationnelles :** Les RDB offrent des fonctionnalités pour gérer les relations entre les tables, effectuer des jointures, filtrer les données et exécuter des requêtes complexes en utilisant SQL.

Exemple: Oracle Database est une base de données relationnelle utilisée dans de nombreuses entreprises pour stocker et interroger des données structurées à grande échelle.

3 - Utilisation

- **Bases de données temporelles :** Les TSDB sont couramment utilisées dans les domaines nécessitant une analyse et une gestion avancées des données temporelles, tels que l'IoT, la finance, la météorologie, la surveillance des capteurs, etc.

Exemple: Une entreprise de télécommunications peut utiliser une base de données temporelle pour stocker et analyser les données de trafic réseau en temps réel.

- **Bases de données relationnelles :** Les RDB sont utilisées dans une large gamme d'applications, notamment les systèmes de gestion de bases de données d'entreprise, les sites web, les applications de commerce électronique, etc.

Exemple: Une entreprise de commerce en ligne peut utiliser une base de données relationnelle pour stocker les informations sur les produits, les commandes des clients et les données de facturation.

2.6.4 Les avantages de l'utilisation d'une TSDB

L'utilisation d'une TSDB (Time Series Database) présente plusieurs avantages significatifs:

- **La possibilité d'analyser de très grandes quantités de données à la fois :** Les TSDB sont conçues pour gérer et analyser des volumes massifs de données temporelles. Elles sont optimisées pour des opérations de lecture et d'écriture rapides, ce qui permet d'analyser efficacement de vastes ensembles de données sans compromettre les performances. Cette capacité est particulièrement importante dans le cas de séries chronologiques où des milliers, voire des millions, de points de données peuvent être collectés à intervalles réguliers.
- **La possibilité de la compression des données :** Les TSDB utilisent souvent des techniques de compression de données pour réduire l'espace de stockage requis. Étant donné que les données temporelles peuvent souvent présenter des motifs et des répétitions, la compression permet de stocker les données de manière plus compacte, ce qui réduit les besoins en espace de stockage. Cela permet d'économiser des ressources et de faciliter la gestion des données à grande échelle.
- **Les TSDB utilisent des interfaces de programmation d'applications API (Application Programming Interface) accessibles en écritures :** Les TSDB offrent des API qui permettent aux développeurs d'écrire et de mettre à jour les données temporelles de manière efficace. Ces API facilitent l'intégration des TSDB dans des applications existantes et permettent de les utiliser pour collecter et stocker en temps réel des données temporelles provenant de diverses sources. L'accès en écriture via des API facilite également l'ajout, la modification et la suppression des données dans la TSDB.

2.6.5 Les applications des séries chronologiques

Les séries chronologiques sont largement utilisées dans de nombreux domaines pour diverses applications. Voici quelques exemples d'applications des séries chronologiques :

1 - Domaine de finance et de commerce

Les données de séries temporelles sont utilisées pour expliquer, prédire et corrélérer le marché financier. Ces données peuvent être examinées afin de donner des prévisions pour les applications qui aident à stabiliser et à atténuer les risques liés aux prix et à la négociation.

Dans l'analyse financière, les analyses des séries chroniques peuvent être utilisées dans le secteur de l'investissement pour le but de la surveillance des performances des taux de titres et les fluctuations dans le temps, de plus la possibilité de prendre les décisions de fabrication, d'achats, allocation et l'optimisation d'utilisation des ressources ainsi que toutes les opérations commerciales.

2 - Domaine de santé

Certains dispositifs ont permis aux médecins d'effectuer une analyse de séries chroniques afin d'établir un diagnostic médical efficace et rapide, ils ont utilisé aussi pour suivre les réponses des patients aux certains traitements, de plus les paramètres de santé des patients doivent être enregistrés avec précision à des intervalles de temps régulières afin de construire une image claire de l'état de santé des patients.

De plus, avec l'avènement des appareils IoT tels que les capteurs portables et les appareils de santé portables, les gens peuvent désormais prendre des mesures régulières de leurs variables de santé au fil du temps avec un minimum d'entrées. Cela conduit à une collecte de données.

3 - Domaine des affaires

Les données de séries chronologiques permettent aux entreprises de prendre des décisions commerciales. Ceci est réalisé lorsque le processus analyse les données passées pour en déduire des événements futurs et révéler des probabilités possibles. Le modèle de données passées est utilisé pour dériver les paramètres suivants :

- **Croissance des affaires** : Pour évaluer la performance financière et commerciale globale et mesurer la croissance, les données de séries chronologiques constituent l'actif le plus approprié et le plus fiable.
- **Estimer la tendance** : les tendances émergentes peuvent être estimées à l'aide de diverses méthodes de séries chronologiques, par exemple, considérez ces méthodes d'analyse des observations de données sur une période de temps pour tenir en compte des augmentations des diminutions des ventes d'un appareil électronique particulier.

- **Dévoiler les modèles saisonniers :** Les points de données enregistrés peuvent révéler des fluctuations et des modèles saisonniers qui facilitent la prévision des données. Les informations obtenues sur les données jouent un rôle clé dans un marché où les prix des matières premières fluctuent de manière saisonnière. Ces données peuvent aider les entreprises à mieux planifier et développer leurs produits.

4 - Domaines météorologiques

Aristote a étudié les conditions météorologiques pour mieux comprendre les causes et les effets observés dans les changements climatiques à long terme. Au fil des jours, les scientifiques ont commencé à enregistrer des données météorologiques sur des instruments tels que des baromètres pour calculer les variables atmosphériques. Les données sont collectées périodiquement et stockées à différents endroits.

Avec le temps, finalement, les prévisions météorologiques ont commencé à apparaître dans les journaux. Avance rapide jusqu'à aujourd'hui, des stations de météorologiques omniprésentes ont été installées dans diverses régions géographiques du monde pour collecter des variables météorologiques précises.

Ces stations disposent d'équipements fonctionnels avancés qui sont interconnectés pour collecter et corrélérer les données météorologiques de différents endroits. Les données pertinentes sont utilisées pour prévoir les conditions météorologiques à tout moment selon les besoins.

5 - Domaine d'astronomie

Les données de séries chronologiques concernent les longueurs d'onde et les intensités de lumière émises par les étoiles, les corps célestes ou les objets. Les astronomes surveillent en permanence ces données de streaming en direct pour détecter les événements cosmiques en temps réel au fur et à mesure qu'ils se produisent.[43]

L'analyse des séries chronologiques a été utilisée pour :

- Découvrez des étoiles lointaines basées sur stellaire les distances.
- Observez les événements cosmiques tels que les supernovæ pour mieux comprendre l'origine de notre univers.

En intégrant l'intelligence artificielle avec les données de séries chronologiques dans ce domaine, le rôle devient la détection et le classement rapide et efficace des objets astronomiques.

2.6.6 Inconvénients des prévisions des séries chronologiques

Bien que les prévisions des séries chronologiques présentent de nombreux avantages, il est important de reconnaître qu'elles ont également quelques inconvénients. Voici certains des principaux inconvénients des prévisions des séries chronologiques :

- **Sensibilité aux données aberrantes :** Les séries chronologiques peuvent être influencées par des valeurs aberrantes ou des événements imprévus. Ces valeurs atypiques peuvent fausser les prévisions et rendre les résultats moins fiables.
- **Incertitude dans les périodes de transition :** Lorsque les données historiques ne fournissent pas suffisamment d'informations sur les périodes de transition, les prévisions peuvent être moins précises. Les changements économiques, politiques ou technologiques importants peuvent rendre difficile la prédiction des futurs schémas de données.
- **Limitations liées aux modèles utilisés :** Les prévisions des séries chronologiques reposent sur des modèles mathématiques et statistiques qui supposent souvent des schémas linéaires ou périodiques dans les données. Cela peut entraîner des imprécisions lorsque les données réelles suivent des schémas plus complexes ou imprévisibles.
- **Dépendance des données historiques :** Les prévisions des séries chronologiques sont basées sur les données historiques disponibles. Si les conditions futures diffèrent considérablement des conditions passées, les prévisions peuvent être moins précises.
- **Effet des facteurs externes :** Les prévisions des séries chronologiques peuvent être influencées par des facteurs externes difficiles à prévoir, tels que les événements politiques, les catastrophes naturelles ou les changements soudains dans le comportement des consommateurs. Ces facteurs peuvent perturber les schémas de données et rendre les prévisions moins fiables.

2.6.7 Les principales TSDB

Il existe plusieurs bases de données des séries chronologiques, les plus importants et leurs classements peuvent être vus dans la figure suivante selon le classement des DB-Engines¹ des systèmes de gestion des bases de données des séries temporelles.

D'après cette figure on peut remarquer le changement historique du classement des TSDBs depuis l'année 2022 jusqu'à l'année 2023.

1. c'est un web site qu'il s'agit d'une liste partielle du classement complet des systèmes de gestion TSDB en fonction de leur popularité des moteurs de recherche, des réseaux sociaux où la mise- à-jours.

Rank			DBMS	Database Model	Score		
Mar 2023	Feb 2023	Mar 2022			Mar 2023	Feb 2023	Mar 2022
1.	1.	1.	InfluxDB +	Time Series, Multi-model	29.15	-0.29	-0.54
2.	2.	2.	Kdb +	Time Series, Multi-model	8.03	+0.73	-0.98
3.	3.	3.	Prometheus	Time Series	7.33	+0.27	+1.00
4.	4.	4.	Graphite	Time Series	6.59	-0.16	+1.11
5.	5.	5.	TimescaleDB	Time Series, Multi-model	4.52	+0.02	+0.06
6.	6.	↑ 7.	RRDtool	Time Series	3.02	+0.02	+0.41
7.	7.	↑ 9.	DolphinDB	Time Series, Multi-model	2.69	-0.16	+1.16
8.	8.	↓ 6.	Apache Druid	Multi-model	2.63	+0.14	-0.63
9.	↑ 10.	↑ 15.	TDengine +	Time Series, Multi-model	2.43	+0.02	+1.67
10.	↓ 9.	↓ 8.	OpenTSDB	Time Series	2.32	-0.12	+0.53
11.	↑ 12.	↑ 12.	QuestDB +	Time Series, Multi-model	1.89	-0.06	+0.73
12.	↓ 11.	↓ 11.	GridDB +	Time Series, Multi-model	1.83	-0.13	+0.49
13.	13.	↓ 10.	Fauna	Multi-model	1.77	-0.12	+0.42
14.	↑ 15.	↑ 24.	M3DB +	Time Series	1.02	-0.07	+0.77
15.	↑ 17.	↑ 18.	VictoriaMetrics +	Time Series	1.01	+0.02	+0.39
16.	↓ 14.	↓ 13.	Amazon Timestream	Time Series	1.00	-0.15	-0.02
17.	↓ 16.	17.	KairosDB	Time Series	0.95	-0.11	+0.30

Figure 2.2 – Classement des bases de données des séries chronologiques d’après le site web DB-Engines ranking.

[44]

1 - InfluxDB:

InfluxDB occupe la première place dans le classement en 2023, et elle était également la meilleure TSDB en 2022. Cette base de données est conçue spécifiquement pour les séries chronologiques et est très appréciée pour ses performances élevées, sa flexibilité et sa facilité d’utilisation. InfluxDB est souvent utilisée dans des domaines tels que l’Internet des objets (IoT), la télémétrie et la surveillance en temps réel.

2 - Kdb :

Kdb se classe deuxième dans la liste. Il s’agit d’une base de données des séries chronologiques relationnelles développée et commercialisée par KX Systems. Kdb est largement utilisée dans le domaine du trading à haute fréquence en raison de ses capacités d’analyse et de traitement de données très rapides. Elle est basée sur le langage de programmation K, qui est spécialement conçu pour les opérations sur les données de séries chronologiques.

3 - Prometheus :

Prometheus est classé troisième dans le classement. Il s’agit d’un système open source approuvé par la Cloud Native Computing Foundation (CNCF). Prometheus est principalement utilisé pour la collecte de métriques spécifiques dans des environnements distribués. Il permet de surveiller et d’alerter sur des événements particuliers, ce qui facilite le diagnostic rapide des problèmes. Prometheus est souvent utilisé dans les architectures basées sur les conteneurs, comme Kubernetes.

4 - Graphite :

Graphite se classe quatrième dans la liste. Il convient de noter que Graphite est un enregistreur de données open source plutôt qu'une base de données de séries chronologiques à part entière. Il est implémenté en Python et prend en charge le JavaScript. Graphite permet de collecter, stocker et afficher des données de séries chronologiques, ce qui facilite la création de visualisations graphiques pour l'analyse et la surveillance des données.

2.7 InfluxDB :

InfluxDB est open source, c'est la plus performante selon DB-Engins, parfaite pour stocker, interroger et analyser en quelques lignes les données des capteurs IOT, les données d'enregistrement de la surveillance, ainsi que la facilité d'utilisation, aucune dépendance externe. La possibilité d'adaptation avec les données régulières et irrégulières et une syntaxe de requête SQL et la possibilité de la pérennité des données.

InfluxDB permet d'autres choses comme la sécurité de données, la gestion des droits d'utilisateurs...

2.7.1 Définition

InfluxDB est une base de données de séries chronologique (TSDB), open source sans schéma, développée par la société InfluxData. Il est écrit en langage de programmation Go, conçue pour gérer les données de séries chronologiques. Il fournit un langage de requête de type SQL. [45]

2.7.2 Principales caractéristiques

InfluxDB offre plusieurs fonctionnalités qui en font une solution puissante pour la gestion des séries chronologiques. Voici quelques-unes des fonctionnalités prises en charge par InfluxDB :

- **Interface Web conviviale :** InfluxDB dispose d'une interface Web intuitive qui permet de créer facilement des tableaux de bord pour visualiser et analyser les données en quelques minutes.
- **Capacité de stockage élevée :** InfluxDB est conçu pour stocker efficacement de grandes quantités de données de séries chronologiques. Il utilise une compression automatique des données pour optimiser l'utilisation de l'espace de stockage.
- **Performance optimisée :** InfluxDB est écrit entièrement en Go, un langage de programmation performant. Il est compilé en un seul binaire, ce qui signifie qu'il n'a pas de dépendances externes et offre de bonnes performances.

- **Requêtes HTTP et API** : InfluxDB prend en charge les requêtes HTTP et offre une API simple mais puissante pour interagir avec la base de données et effectuer des opérations d'écriture et de lecture des données.
- **Langage de requête SQL-like** : InfluxDB utilise un langage de requête expressif et similaire à SQL, ce qui facilite l'interrogation des données agrégées. Cela permet aux utilisateurs de formuler des requêtes complexes pour obtenir des informations précises à partir des données de séries chronologiques.
- **Indexation avec balises** : InfluxDB utilise des balises pour indexer les séries, ce qui permet des requêtes rapides et efficaces en se basant sur les balises spécifiques.
- **Politique de rétention** : InfluxDB offre la possibilité de définir des politiques de rétention pour supprimer automatiquement les anciennes données après un certain laps de temps. Cela permet de gérer l'espace de stockage et de conserver uniquement les données nécessaires.
- **Requêtes continues** : InfluxDB propose des requêtes continues, qui calculent automatiquement des agrégations de données en temps réel. Cela rend les requêtes fréquentes plus efficaces et réduit la quantité de données à traiter.

En combinant ces fonctionnalités, InfluxDB offre une plateforme complète pour la gestion, l'analyse et la visualisation des séries chronologiques, avec des performances élevées, une flexibilité de requête et une facilité d'utilisation.[46]

2.7.3 InfluxDB Cloud

Les entreprises génèrent un flux important de données et doivent être en mesure de les stocker, les interroger et les analyser rapidement pour prendre des décisions commerciales basées sur des données en temps réel. Les solutions basées sur le cloud, telles que InfluxDB Cloud, offrent une plateforme performante, flexible et sans serveur pour gérer les données de séries chronologiques.

InfluxDB Cloud est géré par les experts qui ont créé InfluxDB, et il est sécurisé avec les dernières fonctionnalités de sécurité et correctifs installés et configurés.

InfluxDB Cloud a été conçu pour fonctionner sur différentes plateformes cloud telles qu'Amazon Web Services (AWS), Google Cloud (GC) et Microsoft Azure. Il propose également des services et fonctionnalités pour faciliter le développement, notamment :

- **Les bibliothèques clientes** : InfluxDB Cloud fournit des bibliothèques clientes pour différentes langues de programmation telles qu'Arduino, C, Go, Java, JavaScript, Node.js, Python, R, Ruby, Scala, Swift, PHP et Kotlin. Ces bibliothèques sont facilement accessibles via un nouvel onglet de l'interface utilisateur, ce qui simplifie l'intégration des applications avec InfluxDB.

- **Les APIs** :²InfluxDB Cloud offre un ensemble robuste d'APIs qui permettent un accès programmatique à la plateforme. Ces APIs sont communes à la version open source d'InfluxDB, ce qui signifie que le code peut être écrit une seule fois et exécuté à la fois localement sur une instance open source et sur InfluxDB Cloud.

En résumé, InfluxDB Cloud permet de gérer toute la complexité liée à l'exécution d'une base de données de séries chronologiques. Il offre une solution performante et flexible pour le stockage, l'interrogation et l'analyse des données temporelles, tout en simplifiant le développement et en garantissant la sécurité des données.[47]

2.7.4 Avantages et limitations d'InfluxDB :

Dans cette partie, nous allons explorer les avantages et les limites d'InfluxDB afin de mieux appréhender ses capacités et de comprendre les défis potentiels associés à son utilisation.

1 - Avantages

- **Open source** : InfluxDB est une base de données open source, ce qui signifie qu'elle est accessible gratuitement et que son code source est disponible pour la communauté.
- **Hautes performances** : InfluxDB est optimisé pour les opérations de séries chronologiques, ce qui lui permet de gérer efficacement de grandes quantités de données et d'exécuter des requêtes rapides.
- **Syntaxe d'intégration familière** : InfluxDB utilise le langage de requête InfluxQL, qui est similaire à SQL, ce qui facilite l'intégration et l'utilisation pour ceux qui sont familiers avec les bases de données relationnelles.
- **Gestion de la durée de rétention des données** : InfluxDB offre des fonctionnalités pour gérer la durée de rétention des données, ce qui permet de supprimer automatiquement les données anciennes selon les besoins.
- **Inclus dans l'écosystème InfluxData** : InfluxDB fait partie de l'écosystème InfluxData, qui comprend d'autres outils et solutions pour la collecte, le traitement et la visualisation des données de séries chronologiques.

2 - Limitations d'InfluxDB

- **Difficultés lors de calculs complexes** : InfluxDB peut rencontrer des difficultés lorsqu'il s'agit d'effectuer des calculs complexes ou des agrégations avancées sur les données. Il est plus adapté aux opérations de requêtes simples et rapides.
- **Communauté internet relativement restreinte** : Comparé à d'autres bases de données, InfluxDB peut avoir une communauté internet moins étendue, ce qui

2. Une API (Application Programming Interface) est un ensemble de règles, de protocoles et de définitions de données qui permettent à différentes applications logicielles de communiquer et d'interagir entre elles. En d'autres termes, une API définit les méthodes et les formats d'échange de données qui permettent à deux logiciels distincts de se connecter et de partager des informations de manière standardisée.

peut limiter l'accès à des ressources et à un support en ligne.

- **Certains connecteurs peu performants** : Certains connecteurs d'InfluxDB, notamment le client Java, peuvent présenter des problèmes de performances, ce qui peut affecter les opérations de lecture et d'écriture des données.

2.7.5 Les clients d'InfluxDB

InfluxDB est largement adopté par une variété de clients provenant de divers secteurs et industries. Voici quelques exemples de clients qui utilisent InfluxDB :

- **Des entreprises technologiques renommées** : Des entreprises telles que Cisco, Intel, IBM et Nexmo font confiance à InfluxDB pour la gestion et l'analyse de leurs données de séries temporelles. Ces sociétés tirent parti des fonctionnalités d'InfluxDB pour surveiller et analyser en temps réel leurs systèmes et leurs infrastructures.
- **Domaine de l'IoT** : Des entreprises telles que Philips, Bosch, Particle et Peloton font confiance à InfluxDB pour la collecte et l'analyse des données de leurs produits connectés. InfluxDB leur permet de surveiller les performances, d'optimiser les fonctionnalités et d'améliorer l'expérience utilisateur de leurs produits IoT.
- **Secteur financier** : Les acteurs du secteur financier utilisent InfluxDB pour l'analyse en temps réel des données de marché. InfluxDB leur permet de surveiller les fluctuations des prix, d'identifier les tendances et les modèles, et de prendre des décisions éclairées dans le trading et l'investissement.
- **Secteur de la santé** : Les entreprises du secteur de la santé utilisent InfluxDB pour collecter, stocker et analyser les données médicales. Cela inclut la surveillance des signes vitaux, la gestion des dossiers médicaux électroniques et l'analyse des données de recherche médicale.
- **Fournisseurs de services de télécommunications**: Les fournisseurs de services de télécommunications utilisent InfluxDB pour la surveillance et l'optimisation des performances de leurs réseaux. InfluxDB leur permet de collecter et d'analyser en temps réel les données de trafic, de capacité réseau et de qualité de service.[48]

2.7.6 Architecture d'InfluxDB

Les principales composantes de l'architecture d'InfluxDB open source comprennent :

- 1 - **Serveur InfluxDB** Il s'agit du nœud principal qui exécute InfluxDB. Il est responsable de la gestion des bases de données, de l'écriture et de la lecture des données, de l'exécution des requêtes et du traitement des opérations.
- 2 - **Stockage des données** Les données sont stockées localement sur le serveur InfluxDB. InfluxDB utilise son propre format de stockage optimisé pour les séries temporelles, ce qui permet des opérations de lecture et d'écriture rapides.
- 3 - **API HTTP** InfluxDB expose une API HTTP qui permet aux clients d'envoyer des requêtes pour écrire et lire des données, gérer les bases de données, les mesures, les séries, etc.

- 4 - **SQL-like Query Language** :InfluxDB utilise un langage de requête basé sur le SQL appelé InfluxQL (Influx Query Language) pour interroger les données et effectuer des agrégations, des regroupements, des filtres, etc.
- 5 - **Interface utilisateur**: InfluxDB propose une interface utilisateur web appelée InfluxDB UI pour interagir avec la base de données, afficher les données, exécuter des requêtes et gérer les paramètres.[48]

2.7.7 les concept-clés du modèle de données d’InfluxDB

- 1 - **Measurement** C’est l’unité de base dans InfluxDB, Il s’agit du nom de la mesure dans laquelle les données sont insérées. Chaque mesure représente une série temporelle spécifique. Par exemple, une mesure pourrait représenter les données de température, de pression ou de débit.
- 2 - **Timestamp (Horodatage)** Indique l’instant précis auquel les données ont été collectées ou stockées. Il représente la composante temporelle des données et permet de les organiser chronologiquement.
- 3 - **Tag** Les tags sont des étiquettes associées à chaque point de données dans une mesure. Ils permettent d’ajouter des métadonnées supplémentaires et de faciliter la recherche et le filtrage des données. On a deux types de tags:
 - **Tag Key** Tag Key est le nom de l’attribut ou du descripteur qui identifie la catégorie à laquelle appartient la donnée. Par exemple, dans une mesure de température, un tag key pourrait être ”lieu” pour indiquer l’endroit où la température a été mesurée.
 - **Tag value** Tag Value st la valeur spécifique associée à un tag key pour une donnée. Reprenant l’exemple précédent, si le tag key est ”lieu”, le tag valu pourrait être ”salle de séjour” pour indiquer que la température a été mesurée dans le salon.
- 4 - **Fields (Champs)** : Les champs sont les valeurs numériques des points de données dans une mesure. Ils représentent les données réelles que vous souhaitez stocker et analyser. Par exemple, un champ pourrait être la valeur de température enregistrée à un instant précis. il existe différents fields lesquelles:
 - **Field Key**: c’est le nom du champ (butterflies).
 - **Field value**: c’est la Valeur du champ à un instant t.
 - **Field set (Ensemble de champs)** : Il s’agit d’une collection de champs qui constituent un point de données complet.[48]

2.7.8 Les séries chronologiques et l'intelligence artificielle

Les séries chronologiques et l'IA ont une relation étroite car les techniques d'IA peuvent être utilisées pour analyser et faire des prédictions basées sur des données de séries chronologiques.

Les techniques d'IA, telles que l'apprentissage automatique et l'apprentissage en profondeur, peuvent être appliquées aux données de séries chronologiques pour découvrir des modèles, faire des prédictions et identifier les anomalies ou les valeurs aberrantes dans les données.

L'IA peut également être utilisée pour la prévision de séries chronologiques, ce qui implique de prédire des valeurs futures sur la base de données passées. Cela peut être utile dans une variété de domaines, y compris la finance, l'énergie et les prévisions météorologiques.

Dans l'ensemble, la relation entre les séries chronologiques et l'IA est puissante, et elle a le potentiel d'aider les entreprises et les chercheurs à faire des prédictions et des décisions plus précises sur la base de données. La relation entre les séries chronologiques et l'IA peut être vue sous plusieurs angles, notamment :

- **Perspective statistique :** L'analyse des séries chronologiques est traditionnellement un domaine statistique, où les chercheurs et les praticiens utilisent des méthodes statistiques pour analyser et modéliser les données des séries chronologiques. Les techniques d'IA sont apparues comme une alternative aux méthodes statistiques et se sont révélées efficaces dans de nombreuses applications de TS, en particulier dans les cas où les méthodes statistiques traditionnelles ne sont pas applicables.
- **Perspective d'apprentissage automatique :** les techniques d'IA telles que l'apprentissage automatique et l'apprentissage en profondeur sont de plus en plus appliquées aux données de séries chronologiques. Les modèles d'apprentissage automatique peuvent être formés pour identifier des modèles dans les données de séries chronologiques et faire des prédictions basées sur des données historiques.
- **Perspective de prévision :** La prévision des séries chronologiques est une application importante de l'analyse des séries chronologiques. Les techniques d'IA telles que l'apprentissage automatique et l'apprentissage en profondeur se sont révélées efficaces dans la prévision des séries chronologiques. Ces techniques peuvent être utilisées pour construire des modèles qui prédisent les valeurs futures d'une série chronologique sur la base de données historiques.
- **Perspective de détection d'anomalies :** la détection d'anomalies est une autre application importante de l'analyse des séries chronologiques. Les techniques d'IA peuvent être utilisées pour construire des modèles qui détectent les anomalies dans les données de séries chronologiques. Ces modèles peuvent être utilisés pour détecter des anomalies dans un large éventail d'applications, notamment la cyber sécurité, la détection des fraudes et la maintenance prédictive.

- **Perspective du secteur :** L'analyse des séries chronologiques est utilisée dans un large éventail de secteurs, notamment la finance, la santé, l'énergie et les transports. Les techniques d'IA ont le potentiel d'améliorer la précision de l'analyse des séries chronologiques dans ces industries, conduisant à de meilleures prévisions et à une prise de décision plus éclairée.[16]

2.7.9 Les méthodes de prévision d'une série chronologique

Une série chronologique est composée d'un processus stationnaire, cela signifie que les propriétés statistiques ne varient pas dans le temps, à savoir sa moyenne, sa variance ou encore sa covariance, donc il est plus aisé de faire des prédictions sur une série s'il est possible de supposer que ces futures propriétés statistiques ne seront pas différentes de celles actuellement observées de plus la plupart des modèles de prévisions de séries chronologiques s'appuie sur ces propriétés. En revanche la prévision de séries chronologiques est un type de prévision qui prédit des évènements futurs sur la base de points de données horodatés, les prévisions sont assez larges, parmi lesquelles on a :[49]

- 1 - **Méthode de lissage :** Le lissage des données est une technique statistique qui consiste à supprimer les valeurs aberrantes d'un ensemble de données de séries chronologiques pour rendre un modèle plus visible.[50]
Il peut s'appliquer à toutes les données d'une série temporelle qui contiennent de façon sûre du bruit ou des fluctuations à court terme.
- 2 - **Modèles des régressions :** Cette méthode consiste à ajuster une droite de régression aux données passées et à l'extrapoler pour prédire les valeurs futures. Cela suppose qu'il existe une relation linéaire entre les variables à prédire et les variables explicatives est linéaire, par exemple : Pour prédire les prix des maisons, on peut utiliser la superficie, le nombre de chambres et d'autres variables pour construire un modèle de régression linéaire. Pour prédire le chiffre d'affaires annuel d'une entreprise en utilisant les données des chiffres d'affaires des années précédentes et d'autres facteurs tels que le nombre d'employés, les dépenses publicitaires, etc. Parmi les modèles de régressions on a:
 - **La régression polynomiale :**La régression polynomiale est une extension de la régression linéaire qui permet de modéliser des relations non linéaires entre la variable dépendante et les variables indépendantes. Elle consiste à ajouter des termes polynomiaux d'ordre supérieur à l'équation de la régression. Prenons l'exemple de la prédiction du nombre de ventes d'un produit en fonction du temps. On dispose d'un ensemble de données contenant le nombre de ventes mensuelles du produit au cours des derniers mois. La régression polynomiale est utilisée afin de modéliser la relation entre la variable indépendante qui est le temps, et la variable dépendante qui est le nombre de ventes.

2 - **Modèles ARIMA** : Il est préférable de définir d'abord l'auto régression qui est un modèle de séries chronologique qui utilise les observations des étapes de temps précédentes comme entrée dans une équation de régression pour prédire la valeur à l'étape de temps suivantes.

Les modèles ARIMA décrivent des phénomènes qui évoluent dans le temps, prédisent des valeurs futures et signifient Auto Regressive Integrated Moving Average (en français Moyenne Mobile Auto Régressive Intégrée).

ARIMA peut être déterminée à l'aide de trois paramètres (p, d, q) : p représente le nombre de termes autorégressifs, d le degré des premières différences non saisonnières nécessaires pour la stationnarité et q le nombre d'erreurs de prévision retardées dans la composante de la moyenne mobile. En se basant sur des observations pondérées du passé, il est donc possible de déterminer une prévision correspondante au moyen d'une équation.

Ces modèles peuvent être utilisés pour prédire le taux de chômage mensuel, aussi en Machine Learning dans le domaine de la finance afin de prédire l'évolution du cours des actions, mais aussi en météorologie afin de prédire les températures.

3 - **La méthode de réseaux de neurones** : C'est une technique d'apprentissage automatique qui utilise des modèles de réseau de neurone artificiels pour prédire les valeurs futures d'une série temporelle. Elle consiste à entraîner un modèle de réseau de neurones à apprendre les relations entre les entrées et les sorties d'une série chronologique à partir d'un ensemble de données en utilisant des algorithmes d'apprentissage profond pour trouver des motifs et des relations. Elle peut être utilisée pour prédire les prix de l'immobilier. Les données historiques des prix de l'immobilier sont utilisées pour entraîner un réseau de neurones, qui peut ensuite être utilisé pour le but de la prédiction des prix futurs. En utilisant un modèle de réseaux de neurone artificiels pour la prévision de la demande de produits ou services, la prévision des ventes et la détection des fraudes financières. [16]

2.7.10 Conclusion :

En résumé, ce chapitre nous a permis de découvrir les bases de données de séries chronologiques, qui sont spécialement conçues pour gérer et stocker des données temporelles complexes collectées sur des intervalles fixes. Nous avons également examiné en détail la base de données InfluxDB et ses principales caractéristiques.

De plus, nous avons abordé la relation entre l'intelligence artificielle et les données de séries temporelles, qui jouent un rôle crucial dans de nombreux domaines de notre vie tels que la santé, l'économie, les affaires, et bien d'autres. Ces données temporelles fournissent des informations précieuses pour la prise de décisions et permettent de développer des applications d'intelligence artificielle puissantes et précises.

En comprenant les bases de données de séries chronologiques et leur utilisation dans des domaines clés, nous sommes maintenant prêts à explorer davantage les

possibilités offertes par ces données et à approfondir notre compréhension de leur impact sur les applications d'intelligence artificielle.

Chapter 3

Etude de cas

Introduction

Avant d'entrer dans les détails de notre application IoT dédiée à l'agriculture, il est important de souligner l'importance croissante de la technologie dans ce secteur. L'agriculture moderne est confrontée à des défis tels que la nécessité d'une utilisation efficace des ressources, l'optimisation des rendements, la gestion des risques climatiques et la traçabilité des produits. Les solutions basées sur l'IoT offrent des opportunités sans précédent pour relever ces défis en fournissant des données en temps réel et des analyses approfondies pour prendre des décisions éclairées. Notre application vise à exploiter ces avancées technologiques pour aider les agriculteurs à surveiller et à contrôler leur processus agricole de manière plus efficace, en offrant des informations précieuses pour une gestion proactive et une prise de décision stratégique.

3.1 L'objectif de projet

Notre objectif principal est de concevoir une application qui simule un système complet d'IoT dédié à l'agriculture. Cette application sera capable de générer des données aléatoires qui représentent les mesures collectées par des capteurs ou des appareils IoT réels utilisés dans le domaine agricole. Ces données seront ensuite publiées via le protocole MQTT, réceptionnées, stockées et visualisées grâce à la base de données InfluxDB.

L'objectif global de notre application est de recréer le processus de collecte et d'analyse des données IoT dans un environnement simulé. Nous souhaitons démontrer comment les données peuvent être gérées, stockées et exploitées dans le contexte de l'IoT, en mettant particulièrement l'accent sur l'agriculture. En simulant ce processus, nous voulons mettre en évidence les avantages et les défis liés à l'utilisation de l'IoT pour surveiller et optimiser les opérations agricoles, tout en soulignant les opportunités offertes par l'analyse des données IoT pour une prise de décision éclairée.

Grâce à cette application, nous visons à fournir une illustration concrète des principes et des technologies sous-jacentes aux systèmes IoT utilisés dans le secteur agricole. Nous souhaitons également présenter les différentes étapes du flux de données IoT, depuis la collecte initiale jusqu'à la visualisation finale. En développant cette simulation, nous aurons l'opportunité d'explorer les fonctionnalités spécifiques de l'IoT et de la base de données InfluxDB, tout en mettant en lumière les cas d'utilisation pertinents et les avantages potentiels pour l'agriculture moderne.

En résumé, cette application permet de simuler un système IoT complet pour la collecte, le stockage et la visualisation des données agricoles. Elle offre des fonctionnalités telles que :

- 1 - **La surveillance des conditions météorologiques** : L'application peut simuler la collecte de données météorologiques telles que la température, l'humidité, les précipitations et la vitesse du vent à partir de capteurs répartis dans une ferme ou un champ agricole. Ces données peuvent être générées aléatoirement pour simuler différentes conditions météorologiques. L'objectif est de surveiller et d'analyser les conditions météorologiques pour optimiser les pratiques agricoles, planifier l'irrigation, la fertilisation et la protection des cultures.
- 2 - **Le suivi de la qualité du sol** : cette application peut simuler la collecte de données sur la composition du sol, le pH, les niveaux de nutriments et d'autres paramètres importants pour évaluer la qualité du sol dans différentes zones agricoles. Ces données peuvent aider les agriculteurs à ajuster les pratiques de fertilisation et à optimiser les rendements des cultures.
- 3 - **La gestion de l'irrigation** : L'application peut simuler la collecte de données sur l'humidité du sol à partir de capteurs d'humidité du sol répartis dans les champs. Ces données peuvent être utilisées pour optimiser les schémas d'irrigation en fournissant des informations en temps réel sur les besoins en eau des cultures, ce qui permet de minimiser les pertes d'eau et d'améliorer l'efficacité de l'irrigation.
- 4 - **La surveillance des cultures** : L'application peut simuler la collecte de données sur la croissance des cultures, telles que la hauteur des plantes, l'indice de végétation, la densité de la canopée, etc. Ces données peuvent aider à surveiller la croissance des cultures, à détecter les problèmes potentiels tels que les maladies ou les carences en nutriments, et à prendre des mesures appropriées pour assurer la santé des plantes.

3.2 Pourquoi on a choisi l'agriculture ?

L'IoT joue un rôle essentiel dans le secteur agricole en offrant des solutions innovantes pour optimiser la production et la gestion agricole. Grâce à l'IoT, les agriculteurs peuvent collecter et analyser en temps réel des données précieuses sur des paramètres tels que la météo, l'humidité du sol, la qualité de l'air et la santé des cultures. Ces informations permettent aux agriculteurs de prendre des décisions éclairées pour améliorer l'efficacité des opérations, réduire les coûts, minimiser les pertes et maximiser les rendements. De plus, l'IoT facilite la surveillance à distance des cultures et du bétail, ce qui contribue à une meilleure gestion de la production. Il est évident que l'IoT est étroitement lié à l'industrie agricole, car il permet d'appliquer des méthodes plus intelligentes et plus durables pour répondre aux besoins croissants de l'agriculture moderne.

3.3 Choix des données

Dans le cadre de notre projet, nous nous concentrons sur les mesures suivantes :

3.3.1 Température

La température est un facteur important pour la croissance des plantes et le développement des cultures. Elle peut influencer la germination, la croissance, la floraison et la maturation des cultures.

3.3.2 Humidité

L'humidité relative de l'air et du sol est cruciale pour le maintien d'un environnement favorable à la croissance des plantes. Une humidité inadéquate peut affecter négativement l'absorption des nutriments et la santé globale des cultures.

3.3.3 Précipitations

Les précipitations représentent la quantité de pluie ou de neige tombée sur une zone donnée. Ces données sont essentielles pour la planification de l'irrigation et l'évaluation de l'apport en eau disponible pour les cultures.

3.3.4 Vitesse du vent

La vitesse du vent peut affecter la transpiration des plantes et l'évaporation de l'eau du sol, ce qui a un impact sur les besoins en irrigation. Elle peut également influencer la propagation de maladies ou de ravageurs dans les cultures.

3.3.5 pH du sol

Le pH du sol mesure son acidité ou alcalinité. Il est crucial pour déterminer la disponibilité des nutriments et le développement des racines des plantes. Différentes cultures ont des préférences en termes de pH du sol.

3.3.6 Niveau de nutriments

Les nutriments tels que l'azote, le phosphore et le potassium sont essentiels pour la croissance et la santé des plantes. Mesurer leur niveau dans le sol permet aux agriculteurs de fournir des quantités appropriées d'engrais et de fertilisants.

3.3.7 Niveau d'humidité du sol

Cette donnée indique la quantité d'eau présente dans le sol. Elle est utilisée pour déterminer le moment et la quantité d'irrigation nécessaire pour les cultures.

3.3.8 pH de l'eau

Le pH de l'eau utilisée pour l'irrigation est important, car un pH inadéquat peut affecter la disponibilité des nutriments pour les plantes et causer des problèmes de salinité ou d'alcalinité.

3.3.9 Niveau de sel

Le niveau de sel dans le sol ou l'eau peut être problématique pour certaines cultures, car un excès de sel peut affecter leur croissance et leur santé.

3.3.10 Niveau de contamination

Cette donnée mesure la présence de substances nocives ou de polluants dans le sol, l'eau ou l'air. Elle est cruciale pour garantir la sécurité alimentaire et la durabilité environnementale.

3.4 les outils et les technologies utilisés

Pour la réalisation de notre projet, nous avons adopté une approche polyvalente en faisant appel à une variété d'outils et de technologies. Cette approche nous a permis de profiter des fonctionnalités spécifiques de chaque outil et de maximiser leurs capacités afin de mettre en œuvre notre projet de manière efficace et optimale.

3.4.1 InfluxDB

Nous avons déjà couvert la notion d'InfluxDB dans le deuxième chapitre, où nous avons souligné son importance en tant qu'outil essentiel pour la gestion et l'analyse des données temporelles.

Installation d'InfluxDB :

Nous avons utilisé les commandes suivantes dans PowerShell pour installer InfluxDB après l'avoir téléchargé depuis le site officiel.[51]:



Figure 3.1 – Les commandes d'installation

1 - Téléchargement de l'archive zip d'InfluxDB :

Cette commande utilise la commande `wget` pour télécharger l'archive zip d'InfluxDB à partir de l'URL spécifiée. L'option `-UseBasicParsing` est utilisée pour analyser le contenu de la page Web, et l'option `-OutFile` spécifie le nom du fichier de

sortie.

2 - Extraction de l'archive zip:

Cette commande utilise la commande Expand-Archive pour extraire le contenu de l'archive zip vers le répertoire de destination spécifié. L'option -DestinationPath indique le chemin complet du répertoire où vous souhaitez extraire les fichiers d'InfluxDB.

3.4.2 Mosquitto

Mosquitto est un logiciel de broker MQTT open source largement utilisé qui joue un rôle d'intermédiaire entre les appareils clients MQTT émettant des messages et ceux qui s'y abonnent. En tant qu'implémentation concrète d'un serveur MQTT, Mosquitto propose des fonctionnalités avancées telles que la gestion des sessions, la sécurité, la prise en charge de la qualité de service (QoS) et la rétention des messages.[52]

3.4.3 Le protocole MQTT (Message Queue Telemetry Transport)

Le protocole MQTT est conçu pour permettre des échanges de données de petite taille sur des réseaux présentant des délais importants et une faible bande passante.

Le fonctionnement du protocole repose sur un serveur MQTT appelé broker, auquel les clients se connectent. Les clients peuvent publier (publish) des messages sur des sujets (topics) spécifiques ou s'abonner (subscribe) à ces sujets. Ainsi, les messages publiés sur les topics peuvent être échangés entre les différents clients.

MQTT s'appuie sur le TCP/IP comme base de communication entre machines reposant sur un modèle simple : un serveur (broker) auquel se connectent des clients.

Le client envoie un message au client avec un sujet spécifique (topic), le broker diffuse le sujet à tous les clients abonnés à ce sujet.[53]

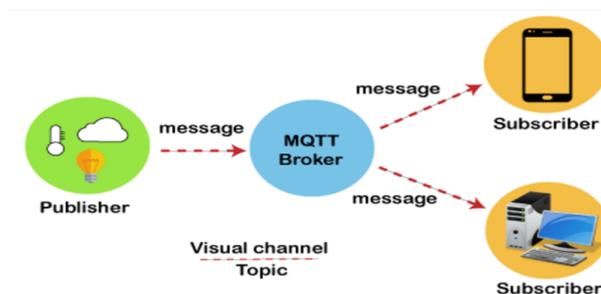


Figure 3.2 – L'architecture du protocole Mqtt

3.4.4 Framework Web Python

Le framework Web Python est une collection de modules ou de packages qui permettent aux développeurs de créer des applications et des services Web sans avoir à gérer les détails de bas niveau tels que les protocoles, les sockets et la gestion des processus ou des threads. Grâce à ce framework, les développeurs peuvent se concentrer sur la création de fonctionnalités et de services pour leur application Web.

Les fonctionnalités que le framework Web Python peut offrir comprennent la gestion des requêtes, l'obtention des paramètres de formulaire, la gestion des cookies et des sessions, la production de réponses dans différents formats, ainsi que le stockage persistant de données et d'autres tâches similaires.

Il existe de nombreux framework Web Python, mais l'un des plus populaires et les plus utiles est le flask.[54]

1 - Flask

Flask est un micro-framework web Python léger qui simplifie la création d'applications web en Python en fournissant des outils et des fonctionnalités utiles. Il offre aux développeurs une grande flexibilité et constitue une option accessible, notamment pour les nouveaux développeurs, car il permet de construire rapidement une application web en utilisant un seul fichier Python. De plus, Flask est extensible et ne nécessite pas une structure de répertoire spécifique ni un code standard complexe avant de commencer.[55]

2 - Les avantages et inconvénients de Flask en Python :

Afin d'avoir une vision complète de Flask en tant que framework Web, il est crucial d'examiner attentivement à la fois ses avantages et ses inconvénients. Cela nous permettra de mieux comprendre les limites de cet outil.

1 - Avantages :

- **Son périmètre** : Il est peu probable de trouver un framework plus léger que Flask en Python. Son installation et son utilisation sont rapides.
- **Flexibilité de Flask** : Flask se distingue par sa grande flexibilité. Il vous permet de résoudre des problèmes et d'intégrer les bibliothèques dont vous avez besoin, en traitant chaque projet individuellement.
- **La courbe d'apprentissage** : Flask est facilement maîtrisable grâce à un tutoriel dédié. Ce framework est indéniablement simple, mais peut également être utilisé pour des projets ambitieux. Flask est une option de choix, tant pour les débutants que pour les professionnels.
- **Open source** : Flask est un framework open source, disponible gratuitement.

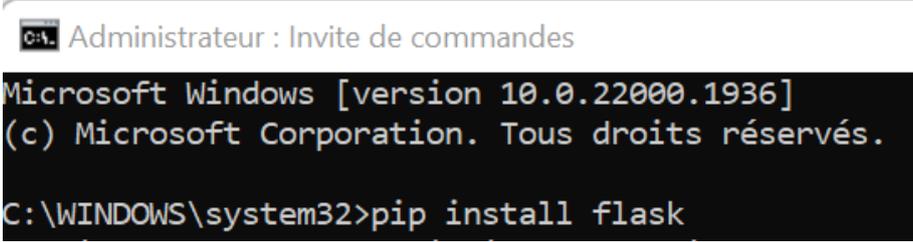
- **La communauté** : Flask rassemble une communauté immense qui offre conseils et soutien aux novices et aux développeurs expérimentés. Les questions et erreurs trouvent rapidement des réponses et des solutions.[56]

2 - Inconvénients :

- **Son périmètre** : En fonction de l'utilisation prévue, le périmètre minimaliste de Flask peut être considéré comme une limite. Chaque fonctionnalité nécessite une installation individuelle, tandis que d'autres frameworks offrent un accès à un plus large éventail de fonctionnalités.
- **Dépendance envers les fournisseurs tiers** : L'utilisation de bibliothèques externes implique toujours un risque d'erreurs, et Flask dépend de ces bibliothèques pour certaines fonctionnalités.
- **La maintenance** : Contrairement à d'autres frameworks qui offrent une maintenance automatique, Flask délègue cette responsabilité à l'utilisateur. Cela permet un contrôle plus élevé, mais nécessite également plus de travail de la part de l'utilisateur.[56]

3 - Installation de Flask

La meilleure façon de procéder à l'installation de Flask est d'utiliser le gestionnaire de packages pip. La commande appropriée est :



```
C:\> Administrateur : Invite de commandes
Microsoft Windows [version 10.0.22000.1936]
(c) Microsoft Corporation. Tous droits réservés.
C:\WINDOWS\system32>pip install flask
```

Figure 3.3 – La commande d'installation de flask

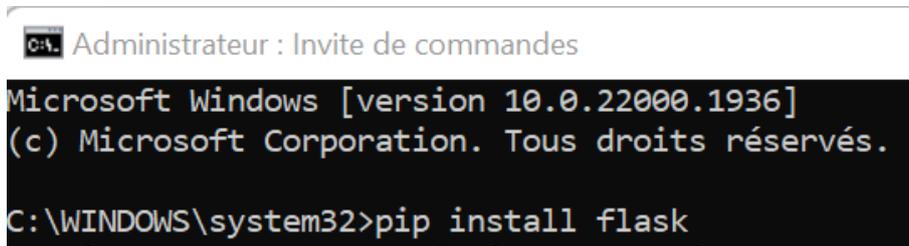
4 - Flask- SocketIO

Est une extension de Flask qui facilite l'utilisation de Socket IO dans une application Flask, elle permet aux applications Flask d'accéder à des communications bidirectionnelles à faible latence entre les clients et le serveur. L'application côté client peut utiliser l'une quelconque des bibliothèques clientes Socket IO en Javascript, Python, C++, Java, Swift ou tout autre client compatible pour établir une connexion permanente avec le serveur.

En utilisant Flask-Socket IO, les développeurs peuvent créer des fonctionnalités de temps réel dans leurs applications Flask, telles que des chats en direct, des tableaux de bord dynamiques, des notifications en temps réel, des jeux multi-joueurs, etc. Cela offre une expérience utilisateur plus interactive et réactive. [57]

5 - Installation de Flask- SocketIO

La commande suivante peut être utilisée pour installer le module Flask-SocketIO.



```
C:\> Administrateur : Invite de commandes
Microsoft Windows [version 10.0.22000.1936]
(c) Microsoft Corporation. Tous droits réservés.
C:\WINDOWS\system32>pip install flask
```

Figure 3.4 – La commande d’installation de flask

3.4.5 Les bibliothèques utilisées :

Dans le but de simplifier le développement de notre application IoT, nous pouvons exploiter plusieurs bibliothèques

A - Bibliothèque JavaScript

Socket.IO:

- définition

Socket.IO est une bibliothèque JavaScript open source qui permet une communication en temps réel, bidirectionnelle et basée sur des événements entre le client et le serveur web. Elle est souvent utilisée dans le développement de jeux en ligne, de salles de discussion, de tableaux de bord en temps réel, de systèmes de suivi et de surveillance, ainsi que d’autres applications web qui nécessitent une communication en temps réel entre le client et le serveur.[58]

- Les caractéristiques :

- La documentation ressemble plus à un didacticiel qui vous aide à créer des applications à partir de zéro.
- Les API uniques de Socket.IO pour les implémentations côté serveur et côté client de plusieurs langages et Framework.
- Énorme communauté, ce qui signifie que vous obtenez une aide gratuite pour de nombreux blocs.
- Il peut créer plusieurs salles et espaces de noms, et les clients peuvent s’abonner à ceux qu’ils préfèrent. [59]

B - Bibliothèque Python

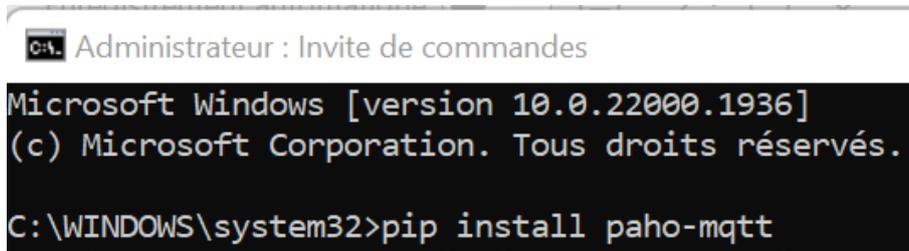
1 - Paho-mqtt :

La librairie Paho-mqtt, développée par Eclipse, propose un client qui permet à des applications de se connecter à un broker MQTT pour publier des messages et s’abonner à des topics afin de recevoir les messages publiés. Il propose aussi des fonctions simples facilitant la publication de messages à un serveur

MQTT.[60]

- Installation de paho-mqtt :

Pour installer le module Paho-mqtt on utilise la commande suivante :



```
C:\> Administrateur : Invite de commandes
Microsoft Windows [version 10.0.22000.1936]
(c) Microsoft Corporation. Tous droits réservés.
C:\WINDOWS\system32>pip install paho-mqtt
```

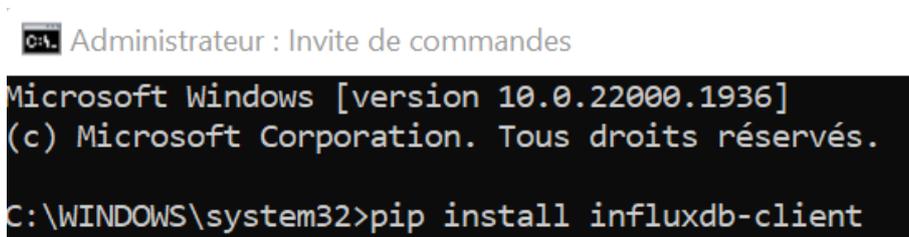
Figure 3.5 – La commande d’installation de paho-mqtt

2 - Influxdb-client :

Bibliothèque Python pour interagir avec une base de données Influx DB. Elle est utilisée pour se connecter au serveur Influx DB, écrire des données et effectuer des requêtes.

- Installation d’influxdb-client

On peut installer cette bibliothèque en utilisant la commande qui suit :



```
C:\> Administrateur : Invite de commandes
Microsoft Windows [version 10.0.22000.1936]
(c) Microsoft Corporation. Tous droits réservés.
C:\WINDOWS\system32>pip install influxdb-client
```

Figure 3.6 – La commande d’installation de flask

3 - La bibliothèque time :

pour la gestion du temps, elle fait référence au temps indépendant du jour (heure, minute, seconde, microseconde).

N.B : Le module time est inclus dans le module utilitaire standard de Python, il n’est donc pas nécessaire de l’installer séparément. Nous pouvons simplement l’importer en utilisant l’instruction d’importation.

4 - La bibliothèque dateTime:

Le module date time fournit des classes permettant de manipuler les dates et les heures.

- **Installation date time** la commande suivante permet l’installation de cette bibliothèque:

```

C:\> Administrateur : Invite de commandes

Microsoft Windows [version 10.0.22000.1936]
(c) Microsoft Corporation. Tous droits réservés.

C:\WINDOWS\system32>pip install DateTime
    
```

Figure 3.7 – La commande d’installation de dateTime

3.4.6 Format JSON :

JSON (JavaScript Object Notation) est un format de données textuelles léger conçu pour l’échange d’informations. Il est facile à générer et à analyser pour les ordinateurs, tandis que sa syntaxe simple et sa structure arborescente le rendent pratique à écrire et à lire pour les humains. JSON est utilisé pour représenter des données structurées de manière similaire à XML, offrant ainsi une alternative flexible et portable pour le stockage et l’échange de données. [61]

3.5 Étapes clés de la réalisation de notre projet

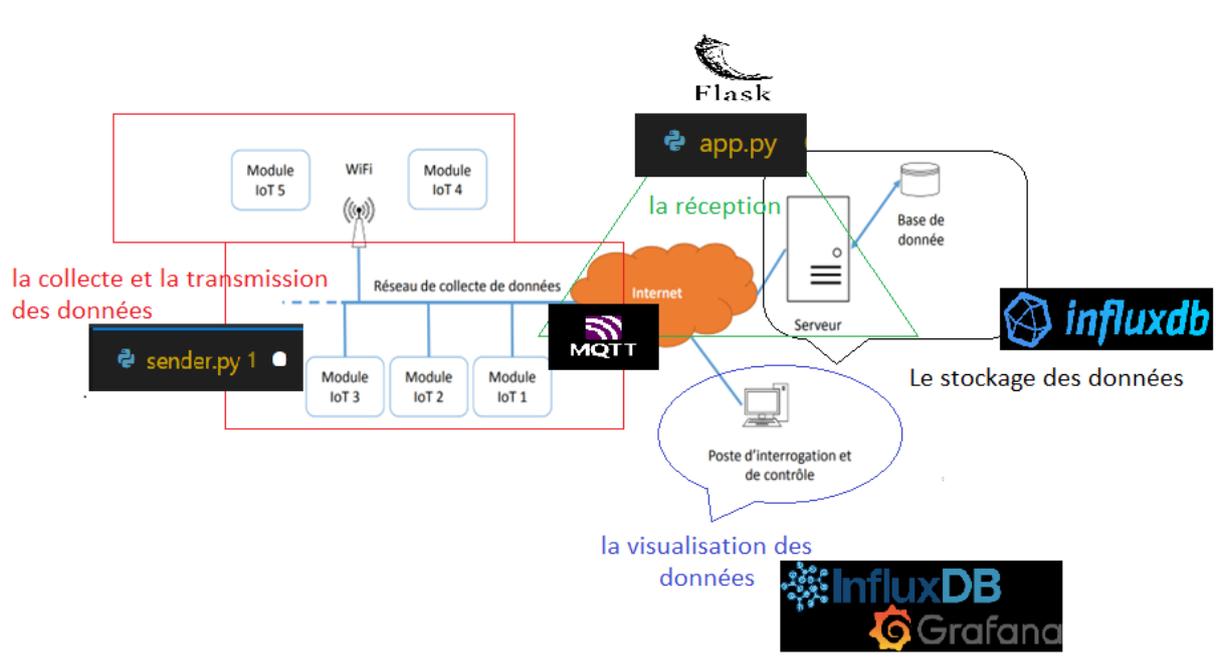


Figure 3.8 – L’architecture globale du projet

1 - Génération de données pour la simulation des capteurs agricoles

Dans cette étape, nous générons des données aléatoires via des scripts python qui joue le rôle des capteurs dans le cas réel tels que la température, l’humidité, les précipitations, la vitesse du vent, etc. Ces données générées aléatoirement simulent les conditions réelles du terrain.

- 2 - **Transmission des données via MQTT** Dans cette étape, les données générées sont transmises via le protocole MQTT pour simuler le processus d'envoi des mesures des capteurs agricoles. Ces données sont publiées sur un topic spécifique du broker MQTT, permettant une communication en temps réel.
- 3 - **Réception des données**
Un serveur MQTT est configuré pour recevoir les messages contenant les données publiées sur le topic spécifié. Les données sont extraites des messages reçus et préparées pour le traitement ultérieur.
- 4 - **Stockage dans une base de données InfluxDB**
Les données agricoles sont ensuite stockées dans une base de données InfluxDB. Cette base de données est spécialement conçue pour gérer efficacement les données de séries chronologiques, ce qui la rend adaptée à notre application IoT agricole.
- 5 - **Visualisation des données**
Une fois les données stockées dans InfluxDB, nous utilisons des outils de visualisation tels que des graphiques ou des tableaux pour présenter les données de manière claire et compréhensible. Cela nous permet d'analyser les tendances, les modèles et les variations des mesures agricoles, facilitant ainsi la prise de décisions éclairées.

3.6 Implémentation du projet

Notre projet implique l'utilisation de deux scripts Python : "sender.py" et "app.py". Ces scripts jouent des rôles essentiels dans la génération, l'envoi, la sauvegarde et la visualisation des données mentionnées précédemment.

Le script "sender.py" est responsable de la génération des données aléatoires qui simulent les mesures provenant des capteurs agricoles. Il utilise des techniques de génération aléatoire pour produire des valeurs de température, d'humidité, de précipitations, de vitesse du vent, etc. Ces données générées sont ensuite prêtes à être envoyées vers le broker MQTT.

D'autre part, le script "app.py" est utilisé pour l'envoi des données générées et leur traitement ultérieur. Il établit une connexion avec le broker MQTT à l'aide de la bibliothèque Paho-mqtt, puis publie les données générées sur un topic spécifique. Une fois les données publiées, le script peut également les récupérer, les stocker dans la base de données InfluxDB et les visualiser via une interface utilisateur.

En résumé, les scripts "sender.py" et "app.py" sont des composants essentiels de notre projet. Ils permettent respectivement la génération des données aléatoires et leur envoi, ainsi que la gestion, le stockage et la visualisation de ces données. Ces scripts jouent un rôle clé dans la simulation d'un système IoT pour l'agriculture en reproduisant le processus de collecte et d'analyse des données provenant de capteurs agricoles.

3.6.1 L'importation des modules :

paho.mqtt.client pour la communication MQTT, **json** pour le formatage des données au format JSON, **time** pour l'horodatage, **datetime** pour la gestion de la date et de l'heure, et **random** pour la génération de nombres aléatoires.

3.6.2 Définition des variables :

Le script définit une variable MQTT-MSG (une liste vide) qui sera utilisée pour stocker les messages MQTT.

3.6.3 Création du client MQTT :

Le script crée un client MQTT en utilisant `mqtt.Client("DataSender")`.

3.6.4 Démarrage du service client MQTT :

Le script démarre le service du client MQTT en appelant `client.loop-start()`. Cela permettra au client de gérer les communications MQTT en arrière-plan.

3.6.5 Connexion au serveur MQTT:

Le script se connecte au serveur MQTT en utilisant `client.connect("127.0.0.1", 1883, 60)`. Il se connecte ici à l'adresse IP locale "127.0.0.1" sur le port 1883 avec un délai de 60 secondes pour établir la connexion.

3.6.6 Définition de fonctions utilitaires :

Le script définit une fonction `GenerateRandom(start, end)` qui génère un nombre aléatoire entre `start` et `end`, arrondi à deux décimales. Il définit également une fonction `RandomData()` qui crée un objet JSON contenant différentes données aléatoires telles que la température, l'humidité, les précipitations, la vitesse du vent, etc.

3.6.7 Boucle principale :

Le script entre ensuite dans une boucle infinie `while(1)` où il publie périodiquement les données aléatoires générées par `RandomData()` en utilisant `client.publish("home/ecole/dat", RandomData())`.

3.7 Fonctionnement et visualisation des données en utilisant InfluxDB :

Pour activer le serveur mosquitto, suivez les étapes suivantes :

- Ouvrez le gestionnaire des tâches de votre système d’exploitation.
- Recherchez l’onglet ou la section intitulée ”Services” ou ”Services système”.
- Parcourez la liste des services disponibles jusqu’à trouver celui associé à mosquitto. Il peut être répertorié sous le nom de ”Mosquitto MQTT Broker” ou quelque chose de similaire.
- Cliquez avec le bouton droit de la souris sur le service mosquitto, puis sélectionnez l’option pour le mettre en action ou l’activer. Selon le gestionnaire des tâches, cela peut être une option telle que ”Démarrer le service” ou ”Activer”. Attendez quelques instants pour permettre au service mosquitto de s’initialiser et de démarrer correctement.

Une fois le serveur mosquitto activé, il sera prêt à gérer les connexions des clients MQTT et à faciliter l’acheminement des messages entre eux. Assurez-vous de laisser le gestionnaire des tâches ouvert pour surveiller l’état du service mosquitto et effectuer d’autres actions si nécessaire, telles que l’arrêt ou la désactivation du service.

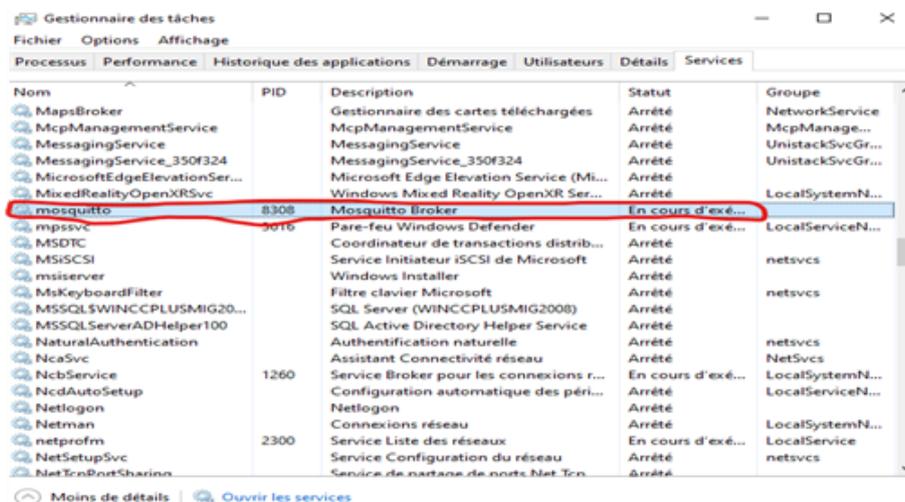


Figure 3.9 – Initialisation du serveur mosquitto

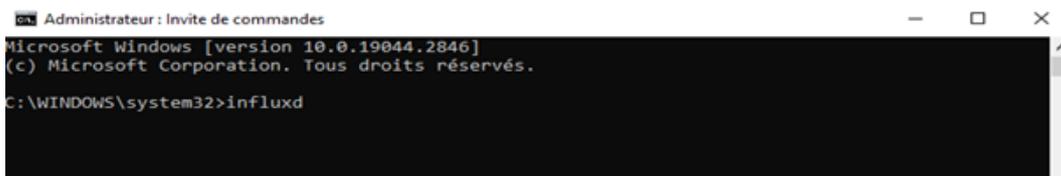
3.8 Démarrage de la base de données influx db

Pour lancer le serveur InfluxDB et exécuter le script Python ”sender.py” ainsi que l’application Flask, nous avons suivi les étapes ci-dessous :

- 1 - Nous avons ouvert l’invite de commande de notre système d’exploitation en tant qu’administrateur. Pour cela, nous avons cherché ”Invite de commande” dans le menu Démarrer, puis cliqué avec le bouton droit de la souris dessus et

sélectionné "Exécuter en tant qu'administrateur".

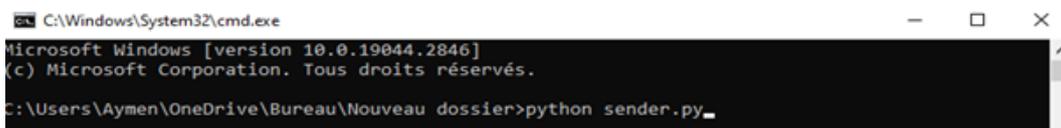
- 2 - Nous avons accédé au répertoire où se trouvent les fichiers "sender.py" et "app.py". Pour ce faire, nous avons utilisé la commande "cd" suivie du chemin d'accès au dossier.
- 3 - Nous avons exécuté le serveur InfluxDB en utilisant la commande "influxd". Avant de lancer cette commande, nous avons vérifié que le serveur était correctement installé sur notre système et que son chemin d'accès était configuré dans les variables d'environnement.



```
Administrateur: invite de commandes
Microsoft Windows [version 10.0.19044.2846]
(c) Microsoft Corporation. Tous droits réservés.
C:\WINDOWS\system32>influxd
```

Figure 3.10 – La commande d'installation de flask

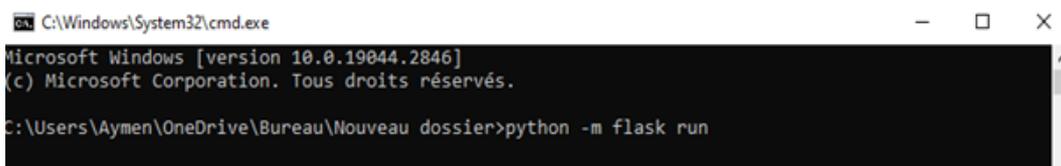
- 4 - Dans une nouvelle fenêtre de l'invite de commande, nous avons exécuté le script Python "sender.py" en utilisant la commande "python" suivie du nom du fichier :



```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19044.2846]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\Aymen\OneDrive\Bureau\Nouveau dossier>python sender.py
```

Figure 3.11 – La commande d'installation de flask

- 5 - Dans une autre fenêtre de l'invite de commande, nous avons exécuté l'application Flask en utilisant la commande "python" suivie du nom du fichier "app.py" :



```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19044.2846]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\Aymen\OneDrive\Bureau\Nouveau dossier>python -m flask run
```

Figure 3.12 – La commande d'installation de flask

Une fois ces commandes exécutées, le serveur InfluxDB sera démarré, le script "sender.py" générera des données et l'application Flask sera lancée. Nous pourrons alors accéder à l'application via un navigateur web en utilisant l'URL <http://127.0.0.1:8086> pour consulter et visualiser les données générées par le script.

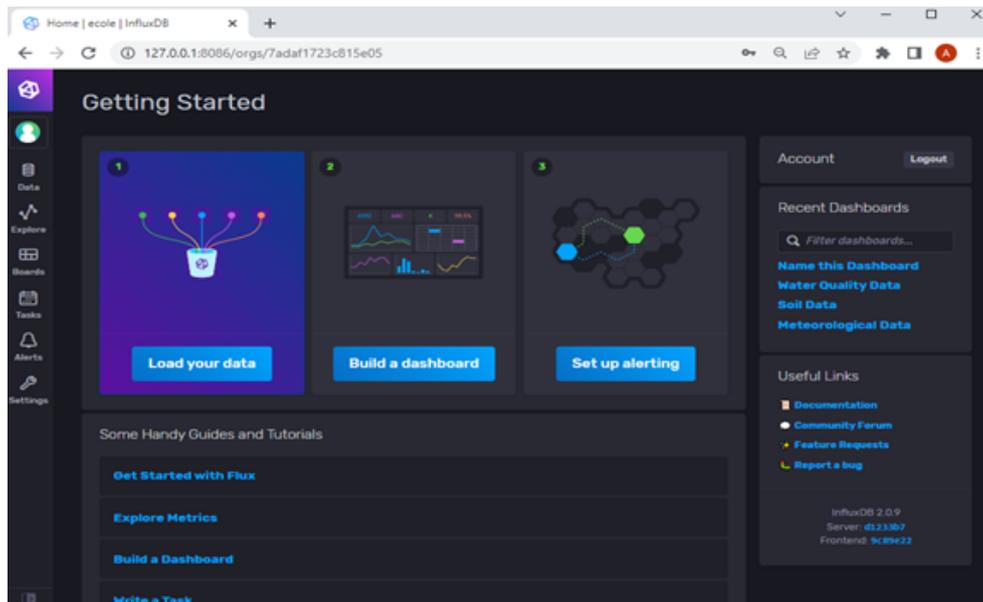


Figure 3.13 – La commande d’installation de flask

3.9 Création du tableau de bord

Pour créer un nouveau tableau de bord (dashboard) dans l’interface d’InfluxDB, nous avons suivi les étapes suivantes :

- 1 - Nous nous sommes assurés d’être connectés à l’interface d’InfluxDB dans notre navigateur web.
- 2 - Dans la partie gauche de l’application, nous avons recherché et cliqué sur ”Dashboards” (tableaux de bord). Cela nous a amenés à la page des tableaux de bord.
- 3 - Sur la page des tableaux de bord, nous avons recherché un bouton ou une option pour créer un nouveau tableau de bord. Cela pouvait être un bouton ”New Dashboard” (nouveau tableau de bord), ”Create Dashboard” (créer un tableau de bord) ou quelque chose de similaire. Nous avons cliqué sur cette option pour commencer la création d’un nouveau tableau de bord.
- 4 - Une fois que nous avons créé le nouveau tableau de bord, nous avons pu commencer à personnaliser son contenu. Pour ajouter une cellule pour afficher les données que nous souhaitions, nous avons recherché une option telle que ”Add Cell” (ajouter une cellule), ”Add Panel” (ajouter un panneau) ou ”Add Visualization” (ajouter une visualisation). Nous avons cliqué sur cette option pour ouvrir un menu ou une interface où nous avons pu configurer la cellule.
- 5 - Dans le menu de configuration de la cellule, nous avons dû spécifier les détails de la visualisation que nous voulions afficher. Cela pouvait inclure le choix du type de graphique (ligne, barre, camembert, etc.), les mesures ou les champs de la base de données à utiliser, les filtres de données, les axes, les couleurs,

etc. Nous avons configuré les paramètres selon nos besoins et préférences.

- 6 - Une fois que nous avons configuré la cellule, nous avons enregistré les modifications ou validé la configuration. Nous avons pu voir la cellule ajoutée à notre tableau de bord, affichant les données souhaitées.

3.10 Les tableaux de bord réalisés

Voici les trois tableaux de bord que nous avons réalisés pour représenter des données importantes dans l'agriculture :

1 - Tableau de bord des conditions météorologiques

Ce tableau de bord affiche les données relatives aux conditions météorologiques essentielles pour l'agriculture. Il comprend des graphiques et des visualisations de la température, de l'humidité, des précipitations et de la vitesse du vent.

Ces informations permettent aux agriculteurs de surveiller de près les conditions météorologiques et d'adapter leurs pratiques agricoles en conséquence.



Figure 3.14 – Tableau de bord des conditions météorologiques

La figure qui suit représente un système IoT avec un capteur virtuel générant des données aléatoires de température.

Les valeurs observées varient entre 0 et 60 degrés Celsius, tandis que la plage de température idéale se situe entre 18 et 25 degrés Celsius (Couleur Verte).



Figure 3.15 – Tableau de bord de température

2 - Tableau de bord de l'irrigation

Ce tableau de bord se concentre sur les données liées à l'irrigation des cultures. Il présente des graphiques montrant les niveaux d'humidité du sol, les quantités d'eau utilisées et les plages d'irrigation recommandées. Ces informations aident les agriculteurs à optimiser l'irrigation en fonction des besoins spécifiques de chaque culture, ce qui contribue à une utilisation efficace des ressources en eau et à une croissance optimale des plantes.

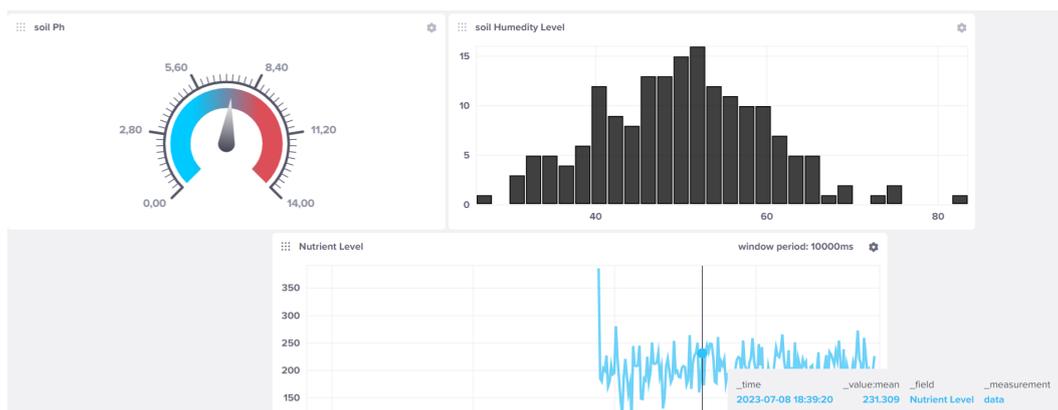


Figure 3.16 – Tableau de bord d'irrigation

la figure suivante met en évidence les mesures de pH du sol, allant de 1 à 14, avec une plage idéale entre 6 et 7,5.

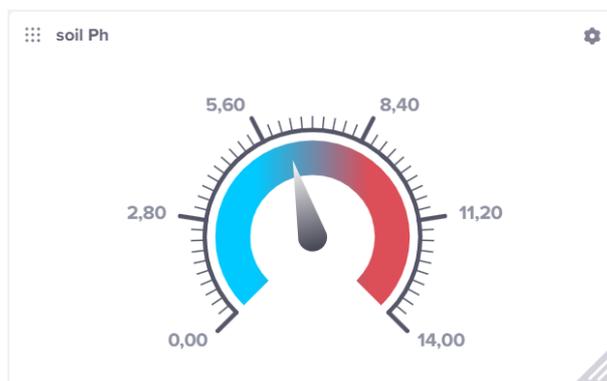


Figure 3.17 – Tableau de bord de pH du sol

Le pH du sol est un paramètre clé pour déterminer sa fertilité et la disponibilité des nutriments pour les plantes.

3 - Tableau de bord de la qualité de l'eau pour la croissance des cultures

Ce tableau de bord met en évidence les données essentielles concernant la qualité de l'eau utilisée pour la croissance des cultures. Il inclut des graphiques montrant des mesures telles que le pH de l'eau, le niveau de sel et le niveau de contamination. Ces informations permettent aux agriculteurs de surveiller attentivement la santé et la progression des cultures, ce qui leur permet de prendre des décisions éclairées concernant la gestion des cultures et les éventuelles interventions nécessaires.

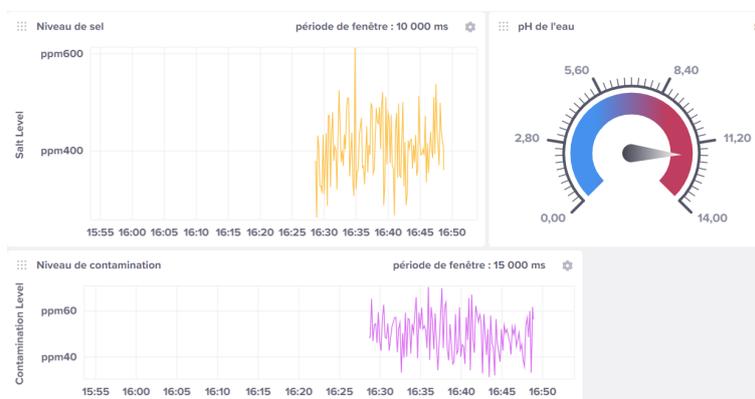


Figure 3.18 – Tableau de bord de la qualité de l'eau pour la croissance des cultures

La figure qui suit présente le niveau de sel dans l'eau

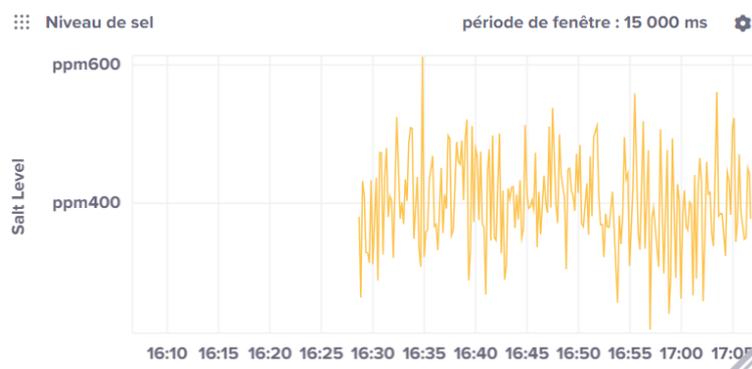


Figure 3.19 – Tableau de bord de la qualité de l'eau pour la croissance des cultures

Nous pouvons observer les niveaux de sel dans l'eau, générés aléatoirement entre 100 et 600 ppm (parties par million). Il est intéressant de noter que les valeurs idéales se situent en dessous de 500 ppm. Maintenir un niveau de sel adéquat est crucial pour diverses applications, telles que l'irrigation et l'aquaculture, afin d'assurer un environnement optimal pour la croissance des plantes ou des organismes aquatiques.

Ces tableaux de bord fournissent aux agriculteurs une visualisation claire et concise des données agricoles essentielles, ce qui facilite la prise de décisions éclairées et l'optimisation des pratiques agricoles.

En outre, InfluxDB nous permet de télécharger nos données sur plusieurs formes, voici un exemple d'un fichier CSV.

_start	_stop	_time	_value	_field	_measurement
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:37:2	82.91	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:37:3	50.014	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:37:4	53.523	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:37:5	43.323	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:38:0	48.229	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:38:1	66.251	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:38:2	63.306	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:38:3	55.196	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:38:4	39.237	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:38:5	50.732	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:39:0	49.551	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:39:1	41.204	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:39:2	56.703	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:39:3	57.05	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:39:4	48.256	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:39:5	53.249	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:40:0	50.985	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:40:1	46.796	ContaminationLevel	data
2023-06-19T12:37:1	2023-06-19T13:37:1	2023-06-19T12:40:2	58.806	ContaminationLevel	data

Figure 3.20 – Une format de données d'un fichier CSV

Ce fichier représente une série de données collecté du niveau de contamination

3.11 Grafana

3.11.1 Définition

Grafana est une plateforme open source, orientée observabilité. Elle est conçue pour surveiller, analyser et visualiser les métriques IT. Elle dispose d'un serveur web intégré écrit en Go, ce qui permet d'y accéder via une API HTTP. Grafana est distribuée sous licence Apache 2.0 et utilise des bases de données de séries temporelles telles que Graphite, InfluxDB ou OpenTSDB pour générer des graphiques et des tableaux de bord. En outre, Grafana offre des fonctionnalités avancées pour la création d'alertes, ce qui en fait un outil essentiel pour la surveillance des systèmes.[62]

3.11.2 Les avantages d'utilisation de Grafana

Grafana présente de nombreux avantages qui en font un choix attrayant pour la visualisation des données et l'analyse.

Voici quelques raisons d'utiliser Grafana :

- **Licence open source** : Grafana est un logiciel open source, ce qui signifie qu'il est gratuit et que vous pouvez le modifier et l'enrichir avec votre propre code. De plus, la communauté de développeurs active contribue à son amélioration continue.
- **Multiplateforme et multi-sources de données** : Grafana peut être installé et utilisé sur différentes plateformes (Linux, Mac, Windows) et est compatible avec une large gamme de sources de données. Vous pouvez visualiser et analyser des données provenant de bases de données telles que Graphite, Prometheus, InfluxDB, Elasticsearch, MySQL, PostgreSQL, etc.
- **Popularité** : Grafana est largement adopté et apprécié par de nombreux développeurs, administrateurs système et entreprises. Des entreprises de renom telles que PayPal, Intel et Stack Overflow font confiance à Grafana pour leurs besoins en matière d'analyse système.
- **Personnalisable** : Grafana offre une grande flexibilité en termes de personnalisation. Vous pouvez créer et modifier des tableaux de bord selon vos préférences et les besoins spécifiques de votre application ou de votre infrastructure. De plus, Grafana permet l'intégration de plugins personnalisés, ce qui vous permet d'étendre ses fonctionnalités de manière encore plus poussée.
- **Communauté active** : Grafana bénéficie d'une communauté active d'utilisateurs et des développeurs qui partagent leurs connaissances, leurs astuces et leurs plugins. Cela facilite l'apprentissage, le support et l'échange d'idées autour de Grafana.[63]

3.11.3 Les inconvénients

Grafana présente quelques inconvénients potentiels :

- **Courbe d'apprentissage**: Il peut falloir du temps pour s'habituer à l'interface et à la configuration de Grafana, surtout pour les débutants.
- **Configuration complexe**: La configuration de Grafana peut être complexe, en particulier lorsqu'il s'agit de connecter différentes sources de données et de créer des tableaux de bord personnalisés.
- **Ressources système**: Grafana peut consommer beaucoup de ressources système, en particulier avec un grand nombre de sources de données, de tableaux de bord ou un trafic élevé.

- **Limitations de la version open source:** La version open source de Grafana a des fonctionnalités limitées par rapport à la version payante.
- **Dépendance aux autres outils:** Grafana est souvent utilisé en conjonction avec d'autres outils, nécessitant une configuration et une maintenance supplémentaires. [63]

3.11.4 A quoi sert Grafana ?

Grafana est une plateforme de surveillance et de visualisation de données.

- Il est utilisé pour la surveillance des systèmes informatiques tels que des serveurs, des réseaux et des applications.
- Grafana permet la visualisation des données d'entreprise provenant de différentes sources telles que des bases de données et des outils de surveillance.
- Il est également utilisé pour la surveillance de l'Internet des objets (IoT) en connectant et en visualisant des données provenant de capteurs et de dispositifs connectés.
- Grafana trouve des applications dans l'analyse des données scientifiques, en aidant à visualiser et à analyser des données complexes.
- Il permet de créer des tableaux de bord personnalisés avec des indicateurs clés de performance (KPI), des graphiques et des jauges.
- Grafana facilite la prise de décisions, la détection de problèmes et l'amélioration des performances grâce à la visualisation et à l'analyse de données provenant de diverses sources.

3.11.5 Démarrage et Accès à l'interface web de Grafana

Après avoir téléchargé Grafana depuis le site officiel [64], vous pouvez procéder à son installation.

Une fois l'installation terminée, vous pouvez démarrer le serveur Grafana en exécutant le fichier correspondant à votre système d'exploitation.

Une fois le serveur Grafana lancé, vous pouvez accéder à son interface web en ouvrant un navigateur et en saisissant l'URL "http://localhost:3000". Cela vous permettra d'accéder à l'interface de Grafana où vous pourrez visualiser, analyser et créer des tableaux de bord personnalisés pour vos données.

3.11.6 Présentation de l'interface de Grafana

Voici une représentation visuelle de l'interface principale de Grafana.

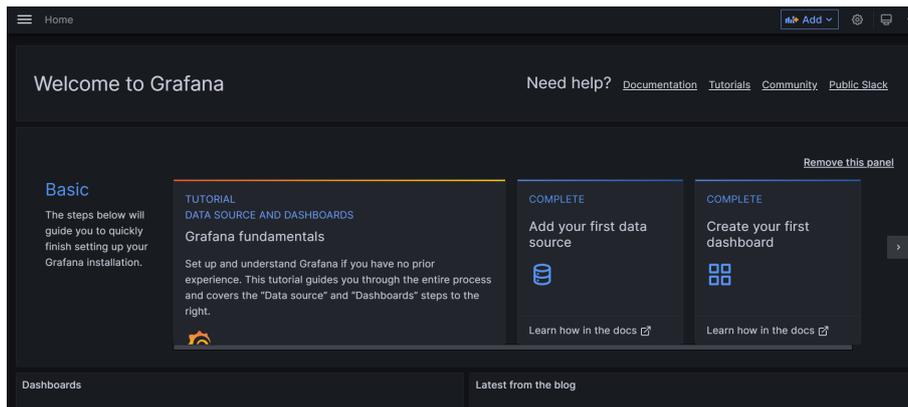


Figure 3.21 – Présentation de l'interface de grafana

3.11.7 La connexion de grafana avec influxdb :

Grafana permet la connexion avec InfluxDB, ce qui vous permet d'utiliser les données associées dans InfluxDB pour créer des visualisations et des tableaux de bord interactifs. Nous commençons alors par configurer la connexion à la source de données InfluxDB.

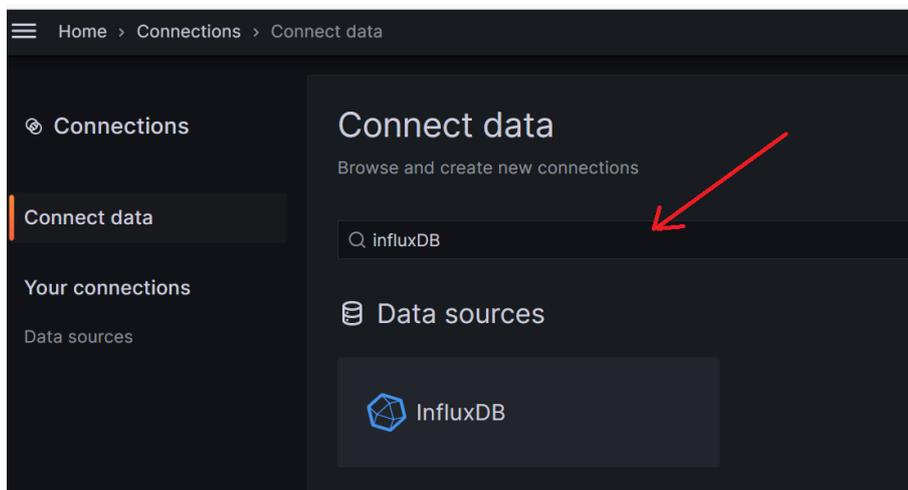


Figure 3.22 – Ajouter influxdb comme une source de données

Dans la section "Paramètres" de la source InfluxDB, nous pouvons configurer les informations d'identification pour la connexion à la source de données InfluxDB. Cela inclut généralement le nom d'utilisateur et le mot de passe, et d'autres paramètres selon vos besoins

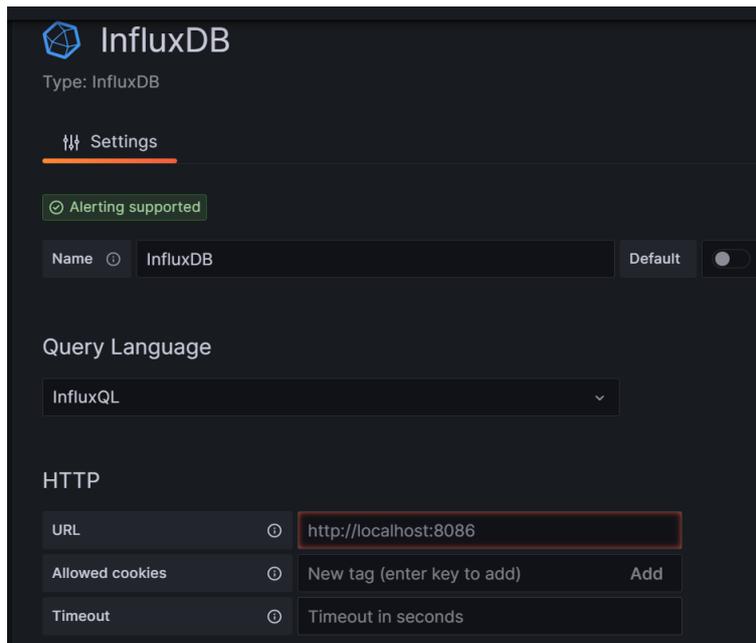


Figure 3.23 – Configuration de la source de données

Dans le pavé InfluxDB Details on mettra le nom de la base de données créée précédemment lors de la configuration d’InfluxDB.

Après avoir configuré et testé la connexion à la source de données InfluxDB dans Grafana, si la connexion est établie avec succès, Grafana affichera généralement un message ou une notification confirmant que la connexion a été réussie.

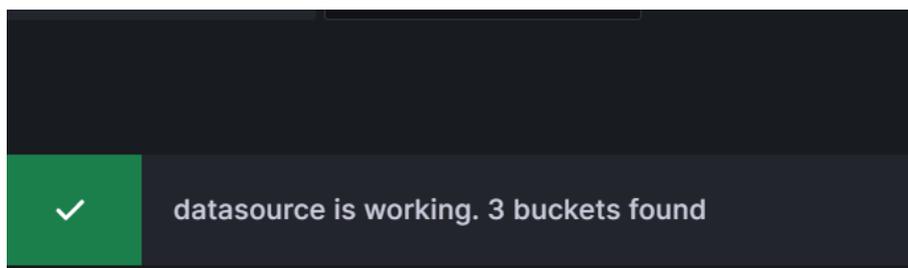


Figure 3.24 – Le test de la connexion à la source de données influxDB est réussi

3.11.8 Création de visualisations

Maintenant que nous avons établi la connexion entre Grafana et InfluxDB, nous sommes prêts à explorer et visualiser nos données. Pour ce faire, nous allons configurer les options de visualisation en spécifiant les métriques, les requêtes et les filtres pertinents afin de récupérer les données de température depuis InfluxDB. Cela nous permettra d’obtenir des visualisations significatives et informatives basées sur ces données spécifiques.

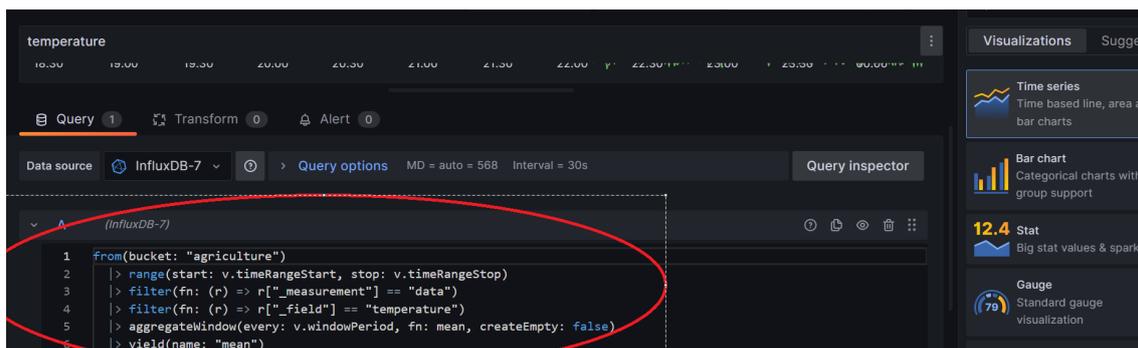


Figure 3.25 – La configuration des données de température depuis influxDB

Après avoir configuré les options de visualisation dans Grafana pour afficher les données de température à partir d’InfluxDB, nous avons personnalisé l’apparence du panneau en ajustant divers paramètres.

Voici le tableau de bord de la température que nous avons réalisé :



Figure 3.26 – Tableau de bord de températures

Ci-dessous quelques exemples réalisés:

Exemple 1

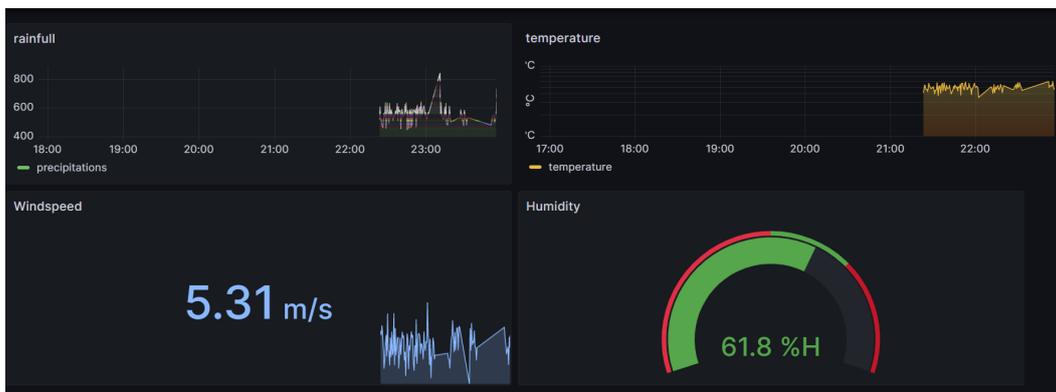


Figure 3.27 – Tableau de bord des données météorologiques

Ce tableau de bord met en évidence les différentes variables météorologiques telles que la température, l’humidité, la vitesse du vent et les précipitations. Les graphiques en temps réel offrent un aperçu instantané de l’évolution des conditions météorologiques.

Exemple 2

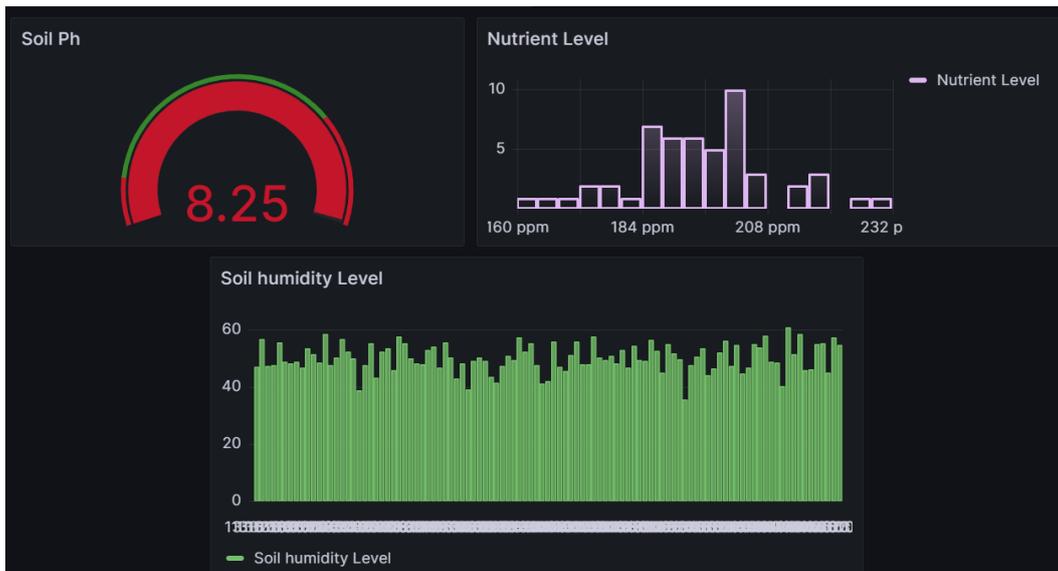


Figure 3.28 – Tableau de bord données de sol

Cette figure présente un tableau de bord affichant des données de sol générées de manière aléatoire. L'aspect visuel de cette visualisation est à la fois attrayant et informatif. Les différentes variables du sol, telles que le pH du sol, le niveau de nutriments et le niveau de l'humidité sont représentées de manière claire et concise.

Exemple 3

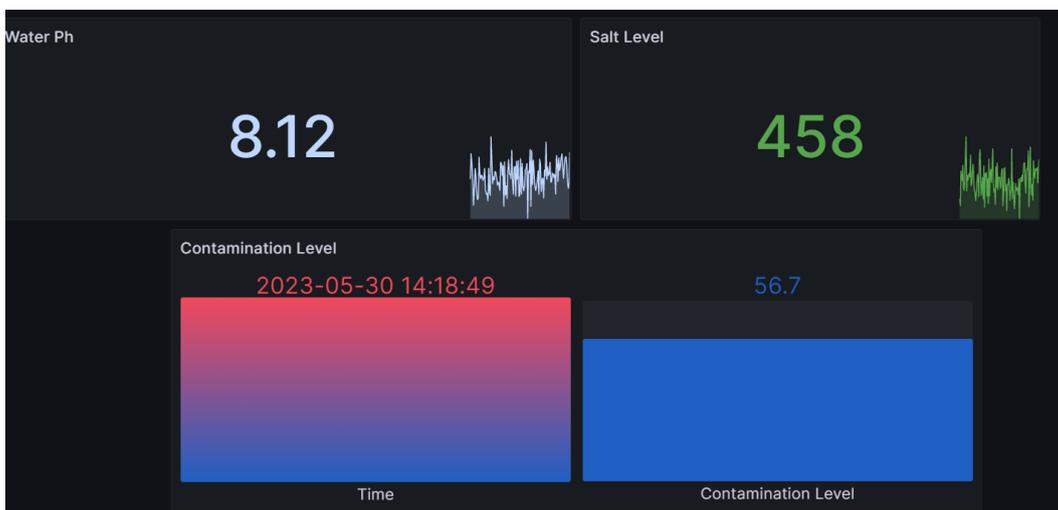


Figure 3.29 – Tableau de bord données sur la qualité de l'eau

Cette image met en évidence un tableau de bord affichant des données de qualité de l'eau, générées de manière aléatoire. L'aspect visuel de cette visualisation est à la fois saisissant et informatif. Les différentes variables de la qualité de l'eau, telles que le pH de l'eau, le niveau de sel et le niveau de contamination sont représentées de manière claire et concise.

3.12 Grafana V.S InfluxDB

le tableau qui suit représente une comparaison entre Grafana et InfluxDB, cette comparaison est basée sur des caractéristiques générales et peut varier en fonction des versions spécifiques de Grafana et InfluxDB, ainsi que des besoins individuels.

Caractéristiques	InfluxDB	Grafana
Type de logiciel	Base de données de séries chronologiques	Outils de visualisation de données
Stockage de données	Optimisé pour les données temporelles	Peut se connecter a différentes sources
Langage de requete	InfluxQL	Requetes SQL, PromQL, etc.
Performance	Ecriture et lecture rapide	Interface utilisateur réactive
Types de graphiques	Graphiques en ligne, diagrammes circulaires, jages, cartes, etc	Divers types de graphiques disponibles
Fonctionnalités avancées	Réplication, haute disponibilité	Exploration de données, filtrage, agrégation, annotation
Intégrations	API pour intégration avec d'autres outils	Prise en charge de nombreuses bases de données
Flexibilité	Spécialisé dans les données temporelles	Peut etre utilisé avec différents types de données

Tableau 3.1 – Tableau comparative entre Grafana et InfluxDB

[16]

3.13 Conclusion

En conclusion de ce chapitre, nous avons porté notre attention sur l'implémentation d'une application IoT destinée au suivi en temps réel du processus agricole. Tout au long de ce chapitre, nous avons exploré en détail les différentes étapes de développement de cette application, en mettant en évidence les technologies clés qui ont été utilisées.

Nous avons commencé par examiner les technologies utilisées pour générer des données aléatoires simulant les mesures provenant des capteurs agricoles. Ensuite, nous avons abordé la publication de ces données via MQTT, la réception et le traitement des messages, ainsi que le stockage des données agricoles dans une base de données InfluxDB.

De plus, nous avons présenté les outils technologiques spécifiques qui ont été employés, tels que Paho-MQTT pour la gestion des connexions MQTT et l'acheminement des messages, ainsi que Flask pour la création de l'application web.

Enfin, nous avons donné un aperçu concret du travail réalisé en présentant les différentes interfaces de l'application web. Ces interfaces permettent la visualisation des données agricoles collectées en temps réel, offrant ainsi aux utilisateurs la possibilité de surveiller et d'analyser les informations essentielles à la prise de décisions agricoles éclairées.

Ce chapitre marque donc une étape cruciale dans le développement de notre application IoT dédiée à l'agriculture. Les technologies mises en œuvre, les outils utilisés et les interfaces créées sont autant de réalisations qui nous rapprochent de notre objectif final : fournir aux agriculteurs un outil puissant et intuitif pour optimiser leurs pratiques agricoles et améliorer leurs rendements.

Conclusion générale

Le présent projet de fin d'études a été consacré à la conception et à la réalisation d'une application IoT pour le suivi en temps réel du processus agricole. Tout au long de ce travail, nous avons abordé différents aspects de l'Internet des Objets (IoT), de la génération de données à la visualisation des informations agricoles, en passant par la collecte, le traitement et le stockage des données.

Dans la première partie de notre travail, nous avons étudié les fondements théoriques de l'IoT et son application dans le domaine de l'agriculture. Nous avons identifié les principaux défis et opportunités liés à l'utilisation de l'IoT dans ce secteur, notamment en ce qui concerne la surveillance des conditions environnementales, la gestion de l'irrigation et la prise de décisions agricoles basées sur des données précises.

Ensuite, nous nous sommes concentrés sur la conception de notre application IoT. Nous avons défini les fonctionnalités clés, les cas d'utilisation et les exigences techniques de notre solution. Nous avons également sélectionné les technologies appropriées pour chaque étape du processus, en mettant l'accent sur la génération de données aléatoires, la communication MQTT, le stockage dans une base de données InfluxDB et la création d'une interface web conviviale.

La mise en œuvre de notre application a nécessité une combinaison de compétences en développement logiciel, en gestion de bases de données et en intégration matérielle. Nous avons développé des scripts Python pour la génération de données aléatoires et la communication MQTT, configuré le serveur InfluxDB pour le stockage des données, et utilisé Flask pour créer une interface web intuitive permettant aux utilisateurs de visualiser les informations agricoles en temps réel.

Lors des tests et des validations, nous avons veillé à ce que notre application réponde aux exigences fonctionnelles et techniques définies précédemment. Nous avons effectué des essais de communication, vérifié l'exactitude des données générées et assuré la stabilité de l'application dans des conditions réelles.

Enfin, nous avons présenté les résultats de notre travail à travers la démonstration de trois tableaux de bord représentant des données importantes dans le domaine de l'agriculture. Ces tableaux de bord fournissent une visualisation claire et précise des mesures agricoles telles que la température, l'humidité et les précipitations, offrant aux utilisateurs une vue d'ensemble de l'état des cultures et facilitant la prise de décisions éclairées.

En conclusion, ce projet de fin d'études nous a permis de mettre en pratique nos connaissances théoriques en matière d'IoT et de les appliquer à un cas concret dans le domaine de l'agriculture. Nous avons acquis une expérience précieuse dans le développement d'applications IoT, en travaillant sur des aspects tels que la

génération de données, la communication MQTT, le stockage dans une base de données et la visualisation des informations.

Ce projet ouvre également la voie à de nombreuses possibilités d'amélioration et d'extension. Des fonctionnalités telles que l'analyse avancée des données, la prédiction des rendements agricoles ou l'intégration de capteurs supplémentaires pourraient être envisagées dans le cadre de travaux futurs.

Ce projet ouvre également la voie à de nombreuses possibilités d'amélioration et d'extension. Voici quelques perspectives pour de futurs travaux :

- 1 - Intégration de l'intelligence artificielle : En utilisant des techniques d'apprentissage automatique et d'intelligence artificielle, il serait possible de développer des modèles prédictifs pour anticiper les rendements agricoles, détecter les maladies des plantes, ou optimiser la gestion des ressources en fonction des conditions environnementales.
- 2 - Exploitation de l'analyse avancée des données : En combinant les données collectées avec des techniques d'analyse avancée, il serait possible de dégager des insights précieux sur les pratiques agricoles, les tendances saisonnières, ou les meilleures stratégies de gestion des cultures.
- 3 - Développement d'une application mobile : En créant une application mobile dédiée, les agriculteurs pourraient accéder aux informations agricoles en temps réel depuis n'importe où, et prendre des décisions rapidement en fonction des alertes et des recommandations fournies par l'application.
- 4 - Collaboration avec des experts agricoles : Pour enrichir davantage l'application, il serait bénéfique de collaborer avec des experts agricoles, des agronomes ou des chercheurs spécialisés dans le domaine de l'agriculture. Leurs connaissances et leur expertise pourraient contribuer à améliorer les fonctionnalités de l'application et à l'adapter aux besoins spécifiques des agriculteurs.

En somme, ce projet représente une étape importante dans l'application de l'IoT à l'agriculture, mais il ouvre également la voie à de nombreuses opportunités d'innovation et de développement. Nous sommes convaincus que les avancées technologiques continueront à transformer le secteur agricole et à soutenir la transition vers une agriculture plus intelligente, plus durable et plus productive.

Enfin, nous espérons que ce travail servira de catalyseur pour de futurs projets et initiatives visant à exploiter pleinement le potentiel de l'IoT dans le domaine de l'agriculture, et à apporter des solutions innovantes pour relever les défis agricoles de notre époque.

Bibliography

- [2] Mr. Fayssal BENDAOU. *cours Internet of Things, Cycle Supérieure Esi-Sba*.
- [24] “The ”Only” Coke Machine on the Internet”. **in**(2014): Retrieved 10 November 2014.
- [25] B. Oscar et I.-N. Service. “”20% des entreprises investissent dans l’Internet des objets””. **in**(10 janv.2016).
- [26] D. Robert. “”Commencer petit pour atteindre des gros succès avec l’Internet des objets””. **in**(janv.2016).
- [27] M. Khezrian K. Munusamy K. Mohebbi S. Ibrahim **and** S. G. H. Tabatabaei. “”A comparative evaluation of semantic web service discovery approaches,”?Proceedings of the 12th International Conference on Information Integration and Web-based Applications Services,” **in**(2010.).
- [28] A. Chikh F. Hadjila **and** A. Belabed. “”Automated discovery of web services: an interface matching approach based on similarity measure,” Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications,” **in**(2010).
- [29] F. Moo-Mena M. Varguez-Moo **and** V. Uc-Cetina. “”Use of classification algorithms for semantic web services discovery”. **in** ” *Journal of Computers, vol. 8, no. 7, pp. 1810-1814: ()*.
- [30] AFOUF Oussama. “Développement d’un Système d’IoT (Internet of Things) dans le cadre de Smart University.” **in**(2020).
- [31] Jean Paul Ezikola Mazoba. “Internet of Things (IoT) and Data Access Control”. **in**(2016).
- [32] BuiNgoc Hai. “Ecoenergetic Software Update Mechanism for Networked IoT Devices”. **in**(2020).
- [33] Rabeb Saad. “Collaborative Model for Internet of Things (IoT)”. **in**(2016).
- [34] HANAFI Imane et YAHIA Samia. “End-to-End IoT System Design for Real-Time Monitoring of Production Processes, Geolocation, and Surveillance”. **in**(2020).
- [35] William S. Cleveland. *Visualizing Data*. Hobart Press, 1993.
- [36] Peter Naur. *Concise Survey of Computer Methods*. Academic Press, 1974.
- [37] David L. Donoho. “Nonlinear Solution of Linear Inverse Problems by Wavelet-Vaguelette Decomposition”. **in***SIAM Journal on Mathematical Analysis: (1995)*.

Bibliography

- [38] C. F. Jeff Wu. *A Course in Statistics with R*. Wiley, 2000.
- [41] Florence Nicolau. “Généralités sur les séries chronologiques”. **in**(2005–2006).

Webographie

- [1] https://fr.wikipedia.org/wiki/Internet_des_objets.
- [3] <https://www.sap.com/products/artificial-intelligence/what-is-iot-internet-of-things.html>.
- [4] <https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et%20infrastructures-iot/>.
- [5] <https://powertech-dz.net/products/single/capteur-de-mouvement-hc-sr501-vente-composants-electronique-blida-algerie-168>.
- [6] https://euro-makers.com/1656-thickbox_default/capteur-de-temperature-et-humidite-arduino-raspberry-dht11-integre.jpg.
- [7] <https://www.amazon.fr/Duracell-Pile-Alcaline-Power-6LR61/dp/B004W7GYGQ>.
- [8] <https://www.eeca.eu/meilleur-capteur-de-courant/>.
- [9] <https://www.aranacorp.com/utilisation-dun-module-rfid-avec-arduino/>.
- [10] <https://www.usinenouvelle.com/expo/verins-pneumatiques-o360.html>.
- [11] <https://www.matooma.com/fr/s-informer/actualites-iot-m2m/objets-connectes-5-criteres-pour-bien-choisir-sa-connectivite>.
- [12] <https://openclassrooms.com/fr/courses/5079046-mettez-en-place-une-architecture-pour-objets-connectes-avec-le-standard-onem2m>.
- [13] <https://www.playhooky.fr/technologie/comment-fonctionne-un-site-web/>.
- [14] <https://www.pickdata.net/news/mqtt-vs-coap-best-iot-protocol>.
- [15] <https://www.pickdata.net/news/mqtt-vs-coap-best-iot-protocol>.
- [16] <https://chat.openai.com/?model=text-davinci-002-render-sha>.
- [17] <https://selectra.info/domotique/actualites/acteurs/protocole-zigbee>.
- [18] <https://iotindustriel.com/iot-iiot/architecture-iot-lessentiel-a-savoir/>.
- [19] <https://www.objetconnecte.com/architecture-reseau-iot>.

- [20] <https://fr.rs-online.com/web/content/blog-discovery/conception-electronique/iot-environnement>.
- [21] <https://www.esilv.fr/objets-connectes-quel-impact-sur-la-vie-privee/>.
- [22] <https://www.lemagit.fr/definition/Intelligence-artificielle-des-objets-AIoT>.
- [23] <https://iotjourney.orange.com/fr-FR/explorer/les-solutions-iot/iot-dans-l-industrie>.
- [39] https://operativeneurosurgery.com/doku.php?id=pooled_data&do=backlink.
- [40] <https://u-next.com/blogs/business-analytics/introduction-time-series-analysis>.
- [42] <https://www.lemagit.fr/definition/Time-series-database-TSDB>.
- [43] <https://geekflare.com/fr/time-series-data/>.
- [44] <https://db-engines.com/en/ranking/time+series+dbms>.
- [45] <https://www.stackhero.io/frfr/services/InfluxDB/documentations/Introduction#qu-est-ce-qu-influxdb>.
- [46] <https://docs.influxdata.com/influxdb/v1.7/>.
- [47] <https://www.influxdata.com/products/influxdb-cloud/>.
- [48] <https://youtu.be/M2IFddZpaJg>.
- [49] <https://www.aquiladata.fr/insights/mieux-comprendre-les-methodes-de-prevision-des-series-chronologiques/>.
- [50] <https://www.influxdata.com/time-series-forecasting-methods/#:~:text=Time%20series%20forecasting%20is%20a,Astronomy>.
- [51] <https://portal.influxdata.com/downloads/>.
- [52] <https://www.aranacorp.com/fr/mettre-en-place-un-serveur-mqtt-avec-mosquitto/>.
- [53] https://qkzk.xyz/docs/nsi/cours_premiere/programmation/outils/libraries_faciles/mqtt/.
- [54] <https://towardsdatascience.com/python-web-framework-a-detailed-list-of-web-frameworks-in-python-1916d3c6222d>.
- [55] <https://python.developpez.com/tutoriel/intro-flask-python3/>.
- [56] <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/framework-flask/>.
- [57] <https://www.delftstack.com/fr/howto/python-flask/flask-websockets/>.
- [58] <https://socket.io/fr/docs/v4/>.
- [59] <https://geekflare.com/fr/best-real-time-applications-software/>.
- [60] https://qkzk.xyz/docs/nsi/cours_premiere/programmation/outils/libraries_faciles/mqtt/.

- [61] <https://flask-socketio.readthedocs.io/en/latest/>.
- [62] <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443882-grafana-gratuit-la-data-visualisation-du-monitoring-it-open-sourc130921/>.
- [63] <https://www.syloe.com/glossaire/grafana/>.
- [64] <https://www.syloe.com/glossaire/grafana/>.

Résumé :

La technologie d'Internet of Things (IoT) a attiré beaucoup d'attention ces dernières années, en particulier dans le domaine de la supervision des données.

L'émergence de l'IoT offre un grand potentiel pour le développement de nouveaux services et applications connectant le monde physique au monde virtuel, un processus qui n'était pas possible auparavant.

L'objectif de ce travail est de mettre en œuvre une conception d'une application spécifique pour le suivi en temps réel basée sur l'IoT en utilisant des outils essentiels tels qu'InfluxDB, Grafana et MQTT.

mots clés: Internet des objets, temps réel, InfluxDB

Abstract :

The Internet of Things (IoT) technology has attracted a lot of attention in recent years, particularly in the field of data monitoring.

The emergence of IoT offers great potential for the development of new services and applications that connect the physical world to the virtual world, a process that was not possible before.

The objective of this work is to implement the design of a specific real-time tracking application based on IoT using essential tools such as InfluxDB, Grafana, and MQTT.

Keywords: Internet of Things, time Series, InfluxDB

الملخص:

جذبت أنترنت الأشياء الكثير من الاهتمام في السنوات الأخيرة، خاصة في مجال مراقبة البيانات.

تتيح ظهور هذه التكنولوجيا فرصًا كبيرة لتطوير خدمات وتطبيقات جديدة تربط العالم الفعلي بالعالم الافتراضي، وهي عملية لم تكن ممكنة في الماضي.

هدف هذا العمل هو تصميم تطبيق محدد لمتابعة البيانات في الوقت الحقيقي باستخدام أدوات أساسية مثل: InfluxDB, Grafana, MQTT.

الكلمات المفتاحية: أنترنت الأشياء, سلسلة زمنية, InfluxDB .