République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

MÉMOIRE DE MASTER

En
Informatique

Spécialité
Administration et Sécurité des Réseaux

Thème

Conception d'une architecture Cloud Hybride pour les systèmes intelligents ambiants

Présenté par : M^lle Rebai Rozina & M. Ouarezki Billel

Soutenue le : 30/06/2024

Devant le jury composé de :

Président	**M^lle. HOCINI**	**Kenza**	**MA**	**U. A/Mira**	**Béjaia.**
Examinateurs	**M^lle. AICHA**	**Azoui**	**MA**	**U. A/Mira**	**Béjaia.**
Encadrant	**M. YAZID**	**Mohand**	**Professeur**	**U. A/Mira**	**Béjaia.**
Co-Encadrant	**M^lle. BENLALA**	**Wissam**	**Doctorante**	**U. A/Mira**	**Béjaia.**

Promotion 2023/2024.

⁕ Remerciements ⁕

Au nom du dieu le clément et le miséricordieux louange à ALLAH le tout puissant.

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidés et bénis par sa volonté durant toute cette période.

Notre profonde gratitude et nos sincères remerciements vont à notre encadrant M. YAZID Mohand et notre co-encadrante Mlle BENLALA Wissam pour leur présence continuelle, leur encouragement et leur patience tout au long de ce travail.

Nous adressons nos remerciements aux membres du jury, devant qui nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire ce mémoire pour juger son contenu.

Nous réservons ici une place particulière pour remercier vivement tous ceux qui, de près ou de loin, nous ont aidés et encouragés tout au long de la réalisation de ce modeste travail.

✳ *Dédicaces* ✳

A ma très chère maman,

Toi qui rêvais tant de me voir réussir et accomplir mes objectifs. Malheureusement, le destin en a décidé autrement et tu nous as quittés trop tôt, sans pouvoir assister à l'aboutissement de tous ces efforts. Mais où que tu sois aujourd'hui, sache que chacune des lignes qui composent ce travail porte une part de toi. Ta force, ton courage et tes sacrifices ont été ma plus grande source de motivation pour aller jusqu'au bout. C'est à toi que je dédie ce modeste projet, qui représente aussi bien ma réussite que la tienne. J'espère qu'il te rend fière de là où tu es, tout comme j'ai toujours été fier d'être ta fille.

À mon cher père, ma sœur Sawsan, mes frères Redouane et Merouane,

Votre soutien indéfectible et votre amour inconditionnel ont été des piliers essentiels dans mon parcours. Je vous dédie également ce travail, fruit de nos efforts conjugués et témoignage de notre lien précieux.

Rebai Rozina

⁕ *Dédicaces* ⁕

Je dédie ce modeste travail en signe de respect, reconnaissance et de remerciement à :

Ma trés chère maman et mon cher père pour leur écoute, leur soutien et leurs encouragements constants durant toutes mes années d'études. Merci infiniment pour votre soutien inestimable.

Mon frère Hamza et ma soeur Sara.

Toute ma famille.

Mes amis.

Ouarezki Billel

Table des matières

Table des figures

Liste des tableaux

Acronymes

API Interface de Programmation Applicative. 10, 12

ARIMA AutoRegressive Integrated Moving Average. 30

AVN Réseaux véhiculaires autonomes. 37

AWS Amazon Web Services. 15

CoAP Constrained Application Protocol. 4

CPU Central Processing Unit. 42, 48

DBSCAN Density-Based Spatial Clustering of Applications with Noise. 31, 33, 68, 70

DDS Data Distribution Service. 4

DirecNet Diabetes Research in Children Network. 36

DL Deep Learning. 27

ECG Électrocardiogramme. 40

EPC Code Électronique de Produit. 2

HTTP Hypertext Transfer Protocol. 4

IaaS Infrastructure-as-a-Service. 15, 16

IDO Internet Des Objets. i, iii, 1, 2, 5–7, 9

IoT Internet of Things. iii, 1, 3–5, 7–11, 17, 21, 27, 35, 36, 38, 39, 41, 45, 46, 49, 57, 81

K-NN K-Nearest Neighbor. 30

LOF Local Outlier Factor. 68, 70

Lora Long Range. 11

MCI Micro Computing Instance. 42

MEC Multi-access Edge Computing. 39–41, 48, 49

MEN Mobile Edge Node. 39

MI-MEC Mining Industry Mobile Edge Computing. 49

MIT Massachusetts Institute of Technology. 2

ML Machine Learning. 32, 55, 56, 67, 69

Mo Mégaoctet. 46

MQTT Message Queuing Telemetry Transport. 4, 35, 59

NB-IoT NarrowBand Internet des Objets. 43–45

NoSQL Not Only SQL. 7

PaaS Platform-as-a-Service. 15

PC Personal Computer. 5

PCA Principal Component Analysis. 30

PDA Patient Data Aggregator. 39

RFID Radio-Frequency Identification. 5

RL Reinforcement Learning. 27

RMSE Root Mean Squared Error. 36

RNN-RBM Recurrent Neural Network-Restricted Boltzmann Machine. 35, 36

SaaS Software-as-a-Service. 16

SQL Structured Query Language. 7

SVM Support Vector Machine. 29, 30, 68, 77

TLS Transport Layer Security. 59

UIT Union Internationale des Télécommunications. 2

VM Virtual Machine. 42

Wi-Fi Wireless Fidelity. 8, 35

XMPP Extensible Messaging and Presence Protoco. 4

Introduction Générale

L'Internet Des Objets (IDO) représente une révolution technologique majeure qui transforme la manière dont les systèmes ambiants fonctionnent. Dans les systèmes ambiants, l'IoT permet une interaction fluide et intelligente entre les dispositifs, créant ainsi des environnements réactifs et adaptatifs. Ces systèmes, souvent intégrés dans des maisons intelligentes, des villes intelligentes, et des environnements industriels, exploitent les capacités de l'IoT pour offrir des solutions innovantes et efficaces [20].
Parallèlement à l'avènement de l'IoT, l'intelligence artificielle et plus particulièrement le machine learning ont connu des avancées significatives. La combinaison de ces technologies ouvre de nouvelles perspectives, notamment dans le domaine des télécommunications, où elles offrent de nouvelles possibilités pour analyser et exploiter efficacement les données massives générées par les dispositifs connectés [39].
L'objectif de ce mémoire est de proposer une architecture cloud hybride pour répondre aux besoins spécifiques dans le domaine des télécommunications. Pour atteindre cet objectif, ce travail se concentrera sur le traitement des données massives générées par les dispositifs IoT, ainsi que sur la mise en œuvre de tests. Ces tests viseront à évaluer les performances des modèles de machine learning appliqués spécifiquement à la prédiction et à la détection d'anomalies dans le domaine des télécommunications, contribuant ainsi à une meilleure gestion des services dans ce secteur et a augmenter les gains de l'entreprise.

Ce mémoire est organisé en trois chapitres :

- ▷ Le premier chapitre traite de l'IoT et les systèmes intelligents ambiants, en présentant les concepts fondamentaux et les applications courantes.
- ▷ Le deuxième divisé en deux parties, explore d'abord les architectures Cloud, Fog et Edge computing, en analysant leurs avantages et leurs défis respectifs, puis présente une revue des travaux connexes dans divers domaines,
- ▷ Le dernier chapitre propose une architecture cloud hybride adaptée, explique les interactions entre les niveaux de l'architecture, les données utilisées, la méthodologie de test synthétique, et enfin, montre les résultats obtenus.

Ce mémoire s'achèvera par une conclusion et des perspectives de recherche induites par notre travail.

Chapitre 1

L'IoT et les Systèmes intelligents ambiants

1.1 Introduction

Avant l'avènement de l'IDO, notre monde était bien moins interconnecté. La majorité des objets du quotidien, comme les électroménagers ou les équipements industriels, fonctionnaient en autonomie sans connexion, rendant difficile la collecte de données et le contrôle à distance. L'IDO a bouleversé ce paradigme en permettant la connexion d'un vaste éventail d'objets à Internet. Cela a été initialement développé par le centre de recherche Auto-ID du Massachusetts Institute of Technology (MIT) dans le but d'identifier de manière unique les produits, aboutissant à la création du Code Électronique de Produit (EPC) commercialisé par EPCglobal. L'objectif initial était de combiner les capacités de communication pour connecter le monde physique au monde numérique. En 2005, l'Union Internationale des Télécommunications (UIT) s'est intéressée aux nouvelles opportunités offertes par cette connectivité des objets, introduisant un nouvel axe de "connectivité à n'importe quoi" complétant les axes "n'importe où" et "n'importe quand". Cependant, cette connectivité soulève de nombreuses questions techniques nécessitant une adaptation du modèle Internet existant. L'IDO, également appelé "réseau d'objets/de choses" ou "Web d'objets", vise à connecter les objets du monde réel pour offrir de nouveaux services, tout en nécessitant une évolution du modèle Internet pour prendre en compte ces nouvelles interactions[19] [16].

1.2 Définitions et notions

1.2.1 Définitions

L'Internet des Objets est un développement infrastructurel mondial visant à connecter à Internet des objets physiques du monde réel (électroménagers, véhicules, machines industrielles, etc.) ainsi que des objets virtuels comme des données et des logiciels. Le principal objectif est de permettre le contrôle à distance de ces objets connectés et la collecte d'informations sur leur environnement ou leur état de fonctionnement. Pour rendre ces objets intelligents et connectés, ils sont équipés de petits dispositifs ayant les capacités suivantes :

- Communiquer et transférer des données via Internet
- Détecter et capter diverses informations de leur entourage (mouvement, température, etc.).
- Stocker les données captées.
- Traiter et analyser ces données.

Grâce à ces capacités, les objets Internet of Things (IoT) peuvent être intégrés dans des applications offrant de nouveaux services pratiques et innovants, tout en garantissant des échanges de données sécurisés[33].

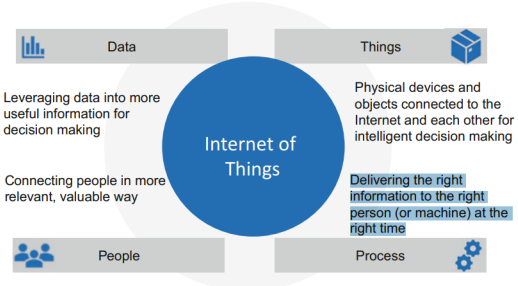

FIGURE 1.1 – La connexion en réseau des personnes, des objets, des données et des processus [17].

1.2.2 Les Protocoles dans l'Internet Des Objets

Des principaux protocoles conçus ou adaptés pour l'écosystème de l'Internet des objets :

1. **Message Queuing Telemetry Transport (MQTT) :** Un protocole de messagerie léger, basé sur un modèle de publication/abonnement, conçu pour les environnements à faible bande passante et haute latence. Il est largement utilisé dans l'IoT pour la communication entre les appareils et le cloud, ainsi que pour la communication machine à machine.

2. **Constrained Application Protocol (CoAP) :** Un protocole de transfert web spécialisé pour les appareils et réseaux contraints, conçu pour la communication machine-à-machine dans l'IoT. Il est basé sur les mêmes principes que Hypertext Transfer Protocol (HTTP) mais optimisé pour les réseaux à faible puissance et sujets aux pertes.

3. **Data Distribution Service (DDS) :** Un protocole de intergiciel pour la communication de données en temps réel, évolutive et haute performance dans les systèmes distribués, souvent utilisé dans les applications industrielles de l'IoT.

4. **Extensible Messaging and Presence Protoco (XMPP) :** Un protocole de communication open source, initialement conçu pour la messagerie instantanée, mais également utilisé dans l'IoT pour la communication entre appareils et entre appareils et serveurs [16].

.

1.2.3 Définition d'un objet dans l'IoT

Les "objets" ou "choses" connectés dans l'IoT sont des objets du quotidien comme des appareils électroniques, des capteurs ou des étiquettes Radio-Frequency Identification (RFID), qui sont connectés via des technologies sans fil ou filaires. Ensemble, ces objets forment un réseau d'atomes (le plus petit élément) connectés à un réseau de bits, constituant ainsi l'IDO. Ces objets connectés peuvent être classés selon leur taille, leur mobilité et s'ils sont animés ou inanimés. L'IDO vise à étendre la connectivité et l'interopérabilité d'objets existants comme les Personal Computer (PC) et téléphones, avec de nouveaux objets connectés par radio, permettant le développement de nouveaux services. Cependant, l'identification, l'adressage et la dénomination de ces objets sont cruciaux, et posent des défis de standardisation et de passage à l'échelle pour gérer potentiellement des milliards d'objets identifiés [19].

1.2.4 Domaines d'application de l'IDO

L'IDO offre un large éventail d'applications dans divers secteurs de la vie quotidienne, transformant les technologies de l'information et de la communication en moteurs d'innovation. Les principaux domaines d'application incluent :

1. **Surveillance de l'environnement :** Permet la détection des risques et la surveillance des phénomènes naturels.
2. **Agriculture intelligente :** Optimise la production agricole grâce à la collecte et au traitement des données.
3. **Transport et logistique :** Améliore la gestion du trafic, le suivi des marchandises et la sécurité des voyageurs.
4. **Villes intelligentes :** Optimise les infrastructures urbaines et améliore la qualité de vie des citoyens.
5. **Maisons et bâtiments intelligents :** Permet d'économiser les ressources et d'améliorer le confort des habitants.

6. **Gestion intelligente des entreprises :** Améliore l'efficacité de production et la gestion des stocks.
7. **Soins de santé :** Facilite la surveillance à distance des patients et améliore la prestation des soins.
8. **Sécurité et surveillance :** Renforce la sécurité dans les espaces publics et privés [22].

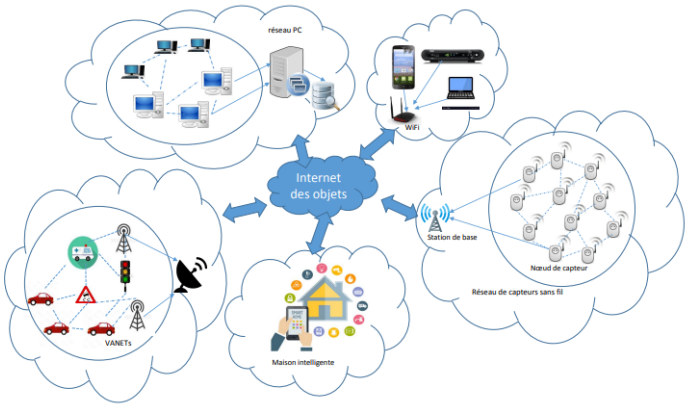

FIGURE 1.2 – Domaines d'application de l'IDO [22].

1.3 L'Internet des Objets à l'ère des Mégadonnées

1.3.1 L'IoT et le Big Data

Les données issues de l'IDO se caractérisent par leurs volumes importants et leur manque d'uniformité, posant de nouveaux défis en matière de gestion des mégadonnées. Jusqu'à présent, la discussion a porté sur ces mégadonnées sans les définir formellement. En effet, les mégadonnées sont un ensemble important de données structurées, non structurées et semi-structurées, ainsi que les résultats de l'analyse de ces données pour en tirer des informations. Comme le souligne Doug Laney, les 3V des mégadonnées sont : le volume (stocker de grandes quantités de données), la vélocité (le taux de génération des données est élevé, donc elles doivent être stockées ou traitées rapidement), et la variété (il existe de nombreux formats possibles pour les données, des données numériques structurées aux e-mails, vidéos, audios, etc.).

Pour exploiter le potentiel de ces données d'IoT, de nouvelles approches techniques sont nécessaires, comme l'analyse automatisée des données, l'apprentissage automatique pour détecter les tendances, l'analyse à la périphérie pour réduire la charge réseau, l'analyse en temps réel pour des résultats immédiats, ou encore l'analyse distribuée pour traiter les très gros volumes. Le stockage de ces données diversifiées fait également appel à des solutions non-relationnelles comme les bases de données NoSQL et de séries chronologiques, plus adaptées que les bases de données SQL traditionnelles. La maîtrise de ces nouvelles techniques de traitement et de stockage des mégadonnées d'IDO permet aux entreprises de tirer parti des précieuses informations stratégiques qu'elles recèlent [19].

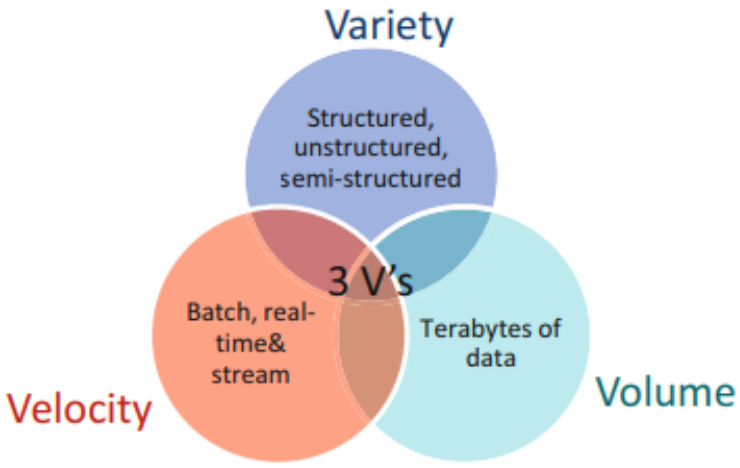

FIGURE 1.3 – Définition du Big Data [17].

1.3.2 Défis

L'IDO apporte de nombreux bénéfices, mais soulève également plusieurs défis importants à relever. Parmi ces principaux défis, on peut citer :

1. **Mise à l'échelle :** Connecter des milliards d'appareils IDO actifs est un défi majeur, nécessitant une adaptation des modèles et technologies de communication actuels.

2. **Hétérogénéité :** L'IDO comporte une multitude d'appa-

reils avec différentes interfaces et protocoles, nécessitant de trouver un moyen commun d'abstraire cette hétérogénéité.

3. **Confidentialité :** Toutes les données collectées doivent être sécurisées et anonymisées lorsque nécessaire.

4. **Responsabilité juridique :** Déterminer qui est responsable en cas de problème avec un algorithme ou une décision automatisée est un défi [17].

1.4 Architectures

L'architecture IoT peut en fait varier considérablement en fonction de la mise en œuvre.Elle doit être suffisamment ouverte avec des protocoles ouverts pour pouvoir prendre en charge plusieurs applications réseau. Dans la majeure partie des cas, elle se compose de 4 blocs constitutifs :

- La scalabilité
- La fonctionnalité
- La disponibilité
- La maintenabilité

Même s'il n'existe pas d'architecture IoT unique universellement acceptée, le format le plus basique et le plus largement accepté est une architecture IoTà quatre couches.

1. **Couche objet :** Il s'agit de la couche physique du matériel composée des dispositifs/objets connectés eux-mêmes (capteurs, actionneurs, équipements) qui collectent des données du monde physique.

2. **Couche réseau :** Cette couche permet la connectivité et le transfert de données entre les dispositifs et les systèmes centraux. Elle utilise divers réseaux de communication et protocoles (Wireless Fidelity (Wi-Fi), Bluetooth, réseaux cellulaires, réseaux étendus de faible puissance,, etc.).

3. **Couche de traitement des données :** Il s'agit de la couche où les données brutes sont traitées, analysées et transformées en informations utiles. Elle inclut le stockage des données, l'intégration, l'analyse, la visualisation, etc. Cette couche s'appuie souvent sur l'informatique en nuage et les technologies du big data.

4. **Couche application :** C'est la couche la plus visible où se trouvent les applications, les interfaces et les services IDO destinés à l'utilisateur final (applications de domotique, systèmes de gestion de bâtiments, applications de suivi de flotte, etc.) [33].

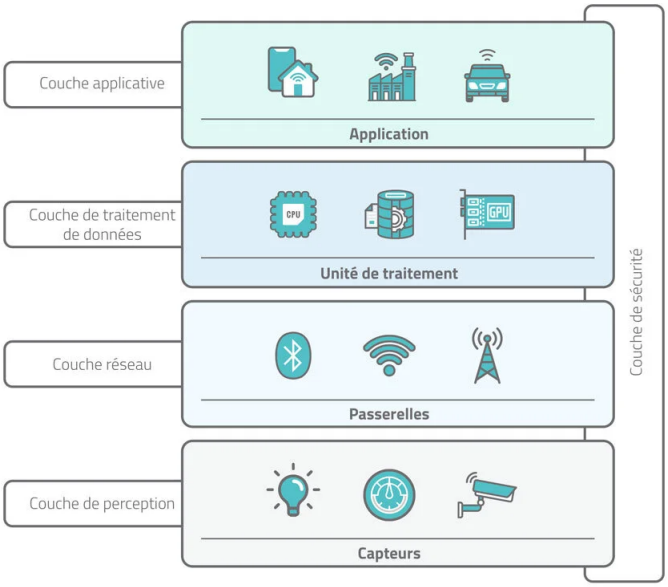

FIGURE 1.4 – Les couches de l'IoT [33].

Dans ce chapitre, on s'intéresse à la couche de traitement de données.

1.4.1 Couche traitement de données

La couche de traitement accumule, stocke et traite les données provenant de la couche précédente. Toutes ces tâches sont généralement traitées via des plateformes IoT et comprennent deux étapes principales.

1. **Étape d'accumulation des données :**

   Les données en temps réel sont capturées via une Interface de Programmation Applicative (API) et mises au repos pour répondre aux exigences des applications non temps réel. L'étape du composant d'accumulation de données fonctionne comme une plaque tournante entre la génération de données basée sur les événements et la consommation de données basée sur les requêtes. Entre autres choses, l'étape définit si les données sont pertinentes pour les besoins de l'entreprise et où elles doivent être placées. Elle enregistre les données dans une large gamme de solutions de stockage, des data lakes capables de contenir des données non structurées telles que des images et des flux vidéo. L'objectif global étant de trier une grande quantité de données diverses et de les stocker de la manière la plus efficace.

2. **Étape d'abstraction des données :**

Ici, la préparation des données est finalisée afin que les applications grand public puissent les utiliser pour générer des informations. L'ensemble du processus comprend les étapes suivantes :

- Combiner des données provenant de différentes sources, à la fois IoT et non-IoT.
- Concilier plusieurs formats de données.

- Agréger les données en un seul endroit ou les rendre accessibles quel que soit leur emplacement grâce à la virtualisation des données.

De même, les données collectées au niveau de la couche application sont reformatées ici pour être envoyées au niveau physique afin que les appareils puissent les comprendre.

Ensemble, les étapes d'accumulation et d'abstraction des données masquent les détails du matériel, améliorant ainsi l'interopérabilité des appareils intelligents[33].

1.4.2 Exemple d'un cas d'utilisation

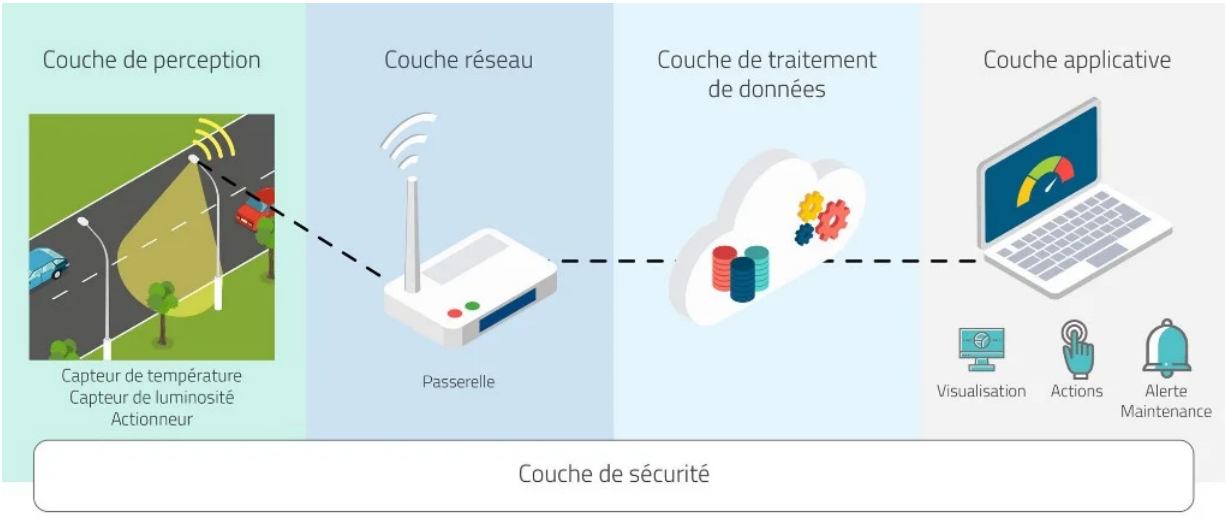

FIGURE 1.5 – Cas d'utilisation IoT [33].

Ceci est un exemple concret pour mieux comprendre et appliquer une architecture IoT optimale. Pour cet exemple, nous avons choisi un lampadaire intelligent. L'objectif étant de monitorer et commander les lampadaires de manière manuelle et automatique.

- **Couche de perception** : Nous disposons d'une lampe intelligente, de capteurs de luminosité et de mouvement ainsi que d'un actionneur. Toutes les données sont remontées via la technologie Long Range (Lora).
- **Couche réseau** : Une gateway est placée pour récupérer les données remontées via les capteurs, elle les traite et les transmet via le réseau cellulaire (5G ou 4G).

- **Couche de traitement de données :** Les données arrivent au serveur Cloud, elles sont accumulées, agrégées et puis mises à disposition sous forme d'une API.
- **Couche applicative** : Elle peut être sous forme Web ou mobile. L'application fait des appels d'API pour récupérer les données relatives au lampadaire (luminosité, état, maintenance...) afin qu'elles soient exploitées. L'application peut allumer ou éteindre les lampadaires automatiquement en fonction de la luminosité ou du mouvement. Et il est aussi possible pour l'utilisateur de commander les lampadaires de manière manuelle. Enfin, l'application permet d'alerter l'utilisateur en cas de défaillance[33].

1.5 Conclusion

L'Internet des objets, en tant qu'évolution de l'Internet actuel, permet d'améliorer considérablement notre mode de vie. Les objets intelligents qui nous entourent peuvent interagir entre eux et avec leurs utilisateurs, de manière à ce que nos activités, nos biens, notre état de santé, nos dépenses, etc. puissent être contrôlés de façon efficace et omniprésente. Mais cela nécessite une bonne architecture cloud pour assurer des communications rapides et fiables entre ces objets. Dans le prochain chapitre, on va expliquer les technologies Cloud Computing, Fog Computing et Edge Computing.

Chapitre 2

Etat de l'art des Architectures Fog/Edge/Cloud Computing

2.1 Introduction

L'Internet des Objets et l'intelligence artificielle génèrent d'immenses quantités de données en continu. Les infrastructures traditionnelles centralisées ne suffisent plus pour gérer efficacement ces données, en raison des besoins croissants en termes de faible latence, de bande passante élevée et de puissance de calcul à la demande. Pour relever ces défis, de nouveaux modèles de gestion des ressources informatiques sont apparus ces dernières années. Le cloud computing a permis d'externaliser le stockage et le traitement des données de manière flexible et évolutive. Cependant, pour répondre aux contraintes de latence et de bande passante des applications critiques, des approches complémentaires comme l'edge computing et le fog computing ont été développées [25]. Ce chapitre se divise en deux parties :
La première partie examine en détail les architectures cloud, fog et edge, ainsi que les solutions hybrides combinant ces paradigmes. Leurs avantages et limites seront analysés en profondeur, offrant une compréhension complète de ces différentes approches. La deuxième partie de ce chapitre fait une revue sur les travaux connexes dans divers domaines en explorant les recherches récentes liées à ces architectures.

Première partie

Les Architectures Cloud, Fog et le Edge computing

2.2 Cloud Computing

2.2.1 Définition

Le cloud computing est un modèle qui permet d'accéder facilement et à tout moment à un ensemble de ressources informatiques partagées et flexibles. Ces ressources incluent : des réseaux, des serveurs, des espaces de stockage, des applications et des services.L'accès à ces ressources se fait via un réseau, généralement Internet. Au lieu de posséder et gérer ses propres ressources informatiques, on les loue de manière flexible auprès d'un fournisseur de cloud selon les besoins. Cela permet d'avoir accès à des capacités informatiques évolutives à la demande, sans les coûts d'investissement et de maintenance d'une infrastructure physique [15].

2.2.2 Modèles de cloud computing

1. **Infrastructure as a Service (Infrastructure-as-a-Service (IaaS)) :** L'IaaS est un modèle de cloud computing où les fournisseurs mettent à disposition des ressources informatiques virtuelles de base (serveurs, stockage, réseau) que les utilisateurs peuvent configurer et gérer eux-mêmes. Les principaux fournisseurs IaaS sont Amazon Web Services (AWS), Microsoft Azure, Google Cloud, etc. Ce modèle offre aux utilisateurs une grande flexibilité et un contrôle élevé sur l'environnement informatique.

2. **Platform as a Service :** Le Platform-as-a-Service (PaaS) est un modèle de cloud computing où les fournisseurs proposent des plateformes informatiques prêtes à l'emploi. Ces plateformes incluent un environnement virtualisé complet avec système d'exploitation, outils de développement, serveurs web, etc. Les utilisateurs peuvent déployer et exécuter leurs propres applications dans cet environnement géré par le fournisseur. Cependant, le niveau de contrôle des utilisa-

teurs est plus limité que dans l'IaaS, car la plateforme est administrée par le prestataire.

3. **Software-as-a-Service (Software-as-a-Service (SaaS)) :** Le SaaS est un modèle de cloud computing où les utilisateurs accèdent à des applications tierces via Internet. Ces applications peuvent être gratuites (Google Docs) ou en abonnement (Dropbox). Les utilisateurs ont très peu de contrôle sur le fonctionnement ou la sécurité de ces applications, qui sont entièrement gérées par le fournisseur cloud. La charge administrative incombe totalement au prestataire de service[14].

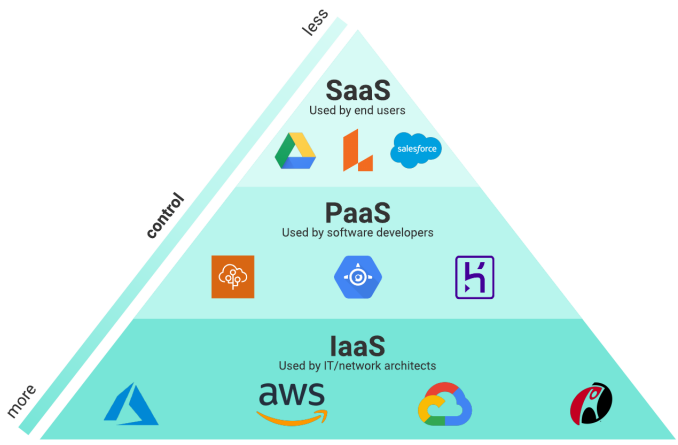

FIGURE 2.1 – Modèles de cloud computing [14].

2.2.3 Architectures :

Comme nous le savons, la technologie du cloud computing est utilisée par les petites et grandes organisations pour stocker des informations dans le cloud et y accéder de n'importe où et à tout moment via une connexion Internet. L'architecture du cloud computing est une combinaison d'architecture orientée services et d'architecture orientée événements. Elle se divise en deux parties principales :

- **Front End (partie cliente) :** C'est la partie visible pour les utilisateurs finaux. Elle inclut les interfaces utilisateur, les

applications et les services qui permettent aux utilisateurs d'interagir avec le cloud.

- **Back End (partie serveur) :** C'est la partie invisible pour les utilisateurs finaux. Elle gère les ressources du cloud, telles que les serveurs, le stockage, les bases de données et les services de traitement [23].

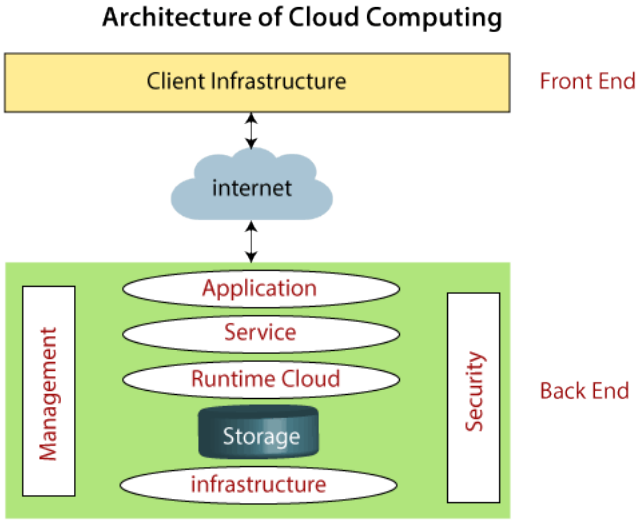

FIGURE 2.2 – Architecture de cloud computing [23].

2.3 Edge Computing

2.3.1 Définition

Le edge computing (L'informatique en périphérie) décentralise les calculs du cloud vers les bords du réseau, au plus près des sources de données. Elle permet de traiter localement les données provenant des objets connectés IoT. Approchant la puissance de calcul des terminaux, elle vise à réduire la latence et la bande passante requise. [15].

2.3.2 Architecture

L'edge computing comprend différents composants qui fonctionnent ensemble pour traiter les données en périphérie. Ces éléments sont les suivants :

- **Couche Dispositif :** contient des capteurs et des contrôleurs

- **Couche Edge :** Sur ce niveau s'effectue ces taches : le traitement et réduction des données, la mise en cache et mise en tampon des données, Réponse de contrôle et la virtualisation.
- **Couche Cloud :** Cette couche fait le traitement des grandes données et l'entreposage des données [28].

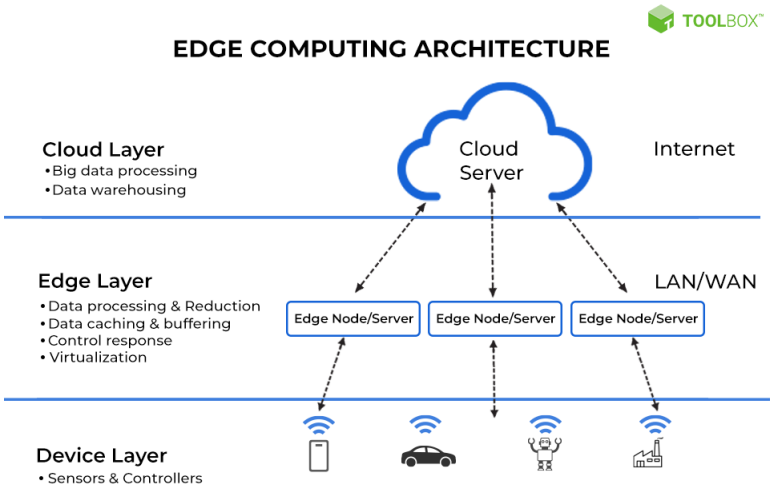

FIGURE 2.3 – Architecture de l'edge computing [28].

2.3.3 Principales caractéristiques de l'edge computing

1. **Des performances de réseau ultra-élevées :** Cloud Edge est une solution révolutionnaire. Il permet aux réseaux d'évoluer rapidement et efficacement à la périphérie grâce à sa combinaison unique de mise à l'échelle linéaire des plans de données utilisateur, de contrôle et de gestion.
2. **Flexibilité du déploiement :** L'edge computing est très flexible et peut être déployée dans n'importe quel environnement, qu'il soit mobile, en cloud ou sur site. Cela permet aux entreprises de déployer des solutions informatiques de pointe de la manière la plus rentable possible.
3. **Expériences différenciées :** L'edge computing permet aux entreprises de différencier les expériences des clients en créant

des engagements personnalisés adaptés aux besoins individuels. Les entreprises peuvent ainsi offrir plus de valeur à leurs clients et les fidéliser davantage.

4. **Sonde virtuelle intégrée et analyse en temps réel :** La sonde virtuelle intégrée et les capacités d'analyse en temps réel de l'edge computing permettent aux entreprises d'obtenir des informations sur le comportement des clients en temps quasi réel. Cela permet aux entreprises de prendre des décisions éclairées sur la meilleure façon de servir leurs clients.

5. **Sécurité :** Alors que la menace de la cybercriminalité augmente, la cybersécurité est l'un des domaines dans lesquels vous ne pouvez pas vous permettre de faire des erreurs. L'edge computing est intrinsèquement sûr, car il élimine la nécessité de transférer des données par l'intermédiaire d'une unité centrale de traitement. Par conséquent, vous serez mieux placé pour protéger vos données contre les attaques malveillantes et réduire le risque de fuite ou de vol de données.

6. **Évolutivité :** Un autre élément clé à prendre en compte lors de l'adoption de nouvelles solutions est la manière dont elles s'adapteront aux besoins de votre entreprise au fur et à mesure de sa croissance. Vous pouvez déployer l'edge computing dans n'importe quel environnement, cela permet aux entreprises d'augmenter ou de réduire facilement leurs effectifs en fonction de leurs besoins [28].

2.4 Fog computing

2.4.1 Définition

l'informatique de brouillard est une architecture informatique distribuée et décentralisée où un grand nombre de dispositifs hétérogènes (sans fil, autonomes, objets, etc.) omniprésents communiquent et coopèrent entre eux ainsi qu'avec le réseau. Ces

dispositifs fournissent des ressources de calcul, stockage, contrôle et réseau de manière collaborative afin d'exécuter des tâches et services dans des environnements isolés C'est une approche intermédiaire entre le cloud centralisé et les appareils terminaux, déployée de manière flexible le long du continuum "cloud aux objets". Les nœuds fog computing peuvent être placés n'importe où entre les dispositifs finaux et l'infrastructure cloud [15].

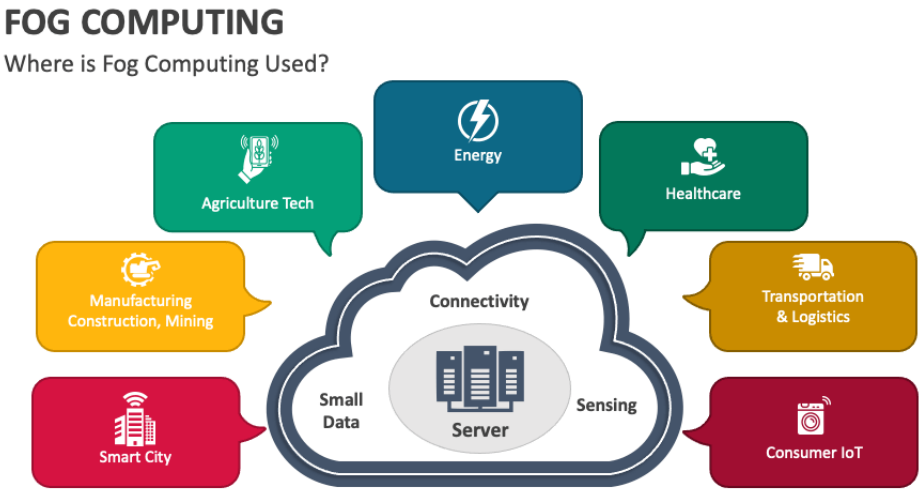

FIGURE 2.4 – Domaines d'utilisation du Fog Computing.

2.4.2 Architecture

2.4.2.1 Architecture à Trois Couches

Cette architecture décentralisée se compose de trois couches distinctes qui collaborent de manière efficace pour fournir des services de proximité et un traitement des données distribué. Les trois couches sont les suivantes :

- **La couche IoT :** La couche IoT comprend divers types d'appareils connectés tels que les capteurs, les véhicules intelligents, les drones, les smartphones et les tablettes. Ces dispositifs sont généralement distribués sur de larges zones géographiques. Leur rôle principal est de collecter des données et de les transmettre aux couches supérieures pour stockage et traitement. Cependant, les appareils dotés de capacités de calcul importantes, comme les smartphones, peuvent également effectuer un traitement local limité avant de faire appel aux couches supérieures.
- **Couche Fog (Brouillard) :** Cette couche intermédiaire représente le noyau de l'architecture Fog computing. Elle comprend une multitude de nœuds Fog, qui sont des éléments réseaux à la fois matériels et logiciels chargés de mettre en œuvre les fonctionnalités du Fog computing.
- **Couche Cloud :** La couche Cloud est composée de l'infrastructure centralisée du cloud computing, comprenant de nombreux serveurs avec de grandes capacités de calcul et de stockage. Elle fournit divers services. Cependant, contrairement à l'architecture cloud traditionnelle, dans l'architecture Fog, certains calculs et services peuvent être migrés de manière efficace de la couche Cloud vers la couche Fog intermédiaire. Ceci permet de réduire la charge sur les ressources du cloud et d'augmenter l'efficacité globale du système [15].

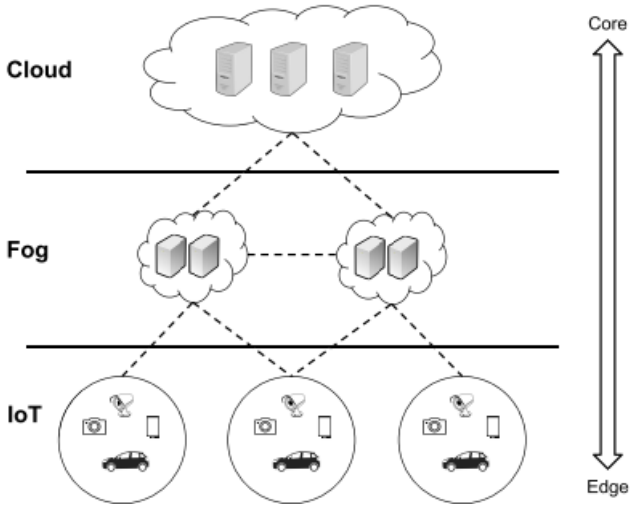

FIGURE 2.5 – Architecture à 3 niveaux du Fog computing [15].

2.4.2.2 Architecture N-TIER OPENFOG

L'architecture N-tier OpenFog structure la couche Fog en plusieurs niveaux (tiers) de nœuds Fog, entre les objets et le cloud. Les niveaux inférieurs se concentrent sur la collecte et la normalisation des données. Les niveaux intermédiaires filtrent, compriment et transforment les données. Le niveau supérieur, proche du cloud, agrège les données et en extrait des connaissances. Plus un nœud Fog est éloigné des terminaux, plus ses capacités de calcul et d'intelligence sont importantes, permettant un raffinement croissant des données. Le nombre de niveaux dépend des exigences du scénario (nombre de terminaux, charge, latence, etc). Les nœuds Fog à chaque niveau peuvent former un maillage, communiquant horizontalement et verticalement, pour la résilience, la tolérance aux pannes, l'équilibrage de charge, etc. Cette architecture vise à guider le déploiement de l'informatique Fog de manière adaptée aux besoins spécifiques du scénario [15].

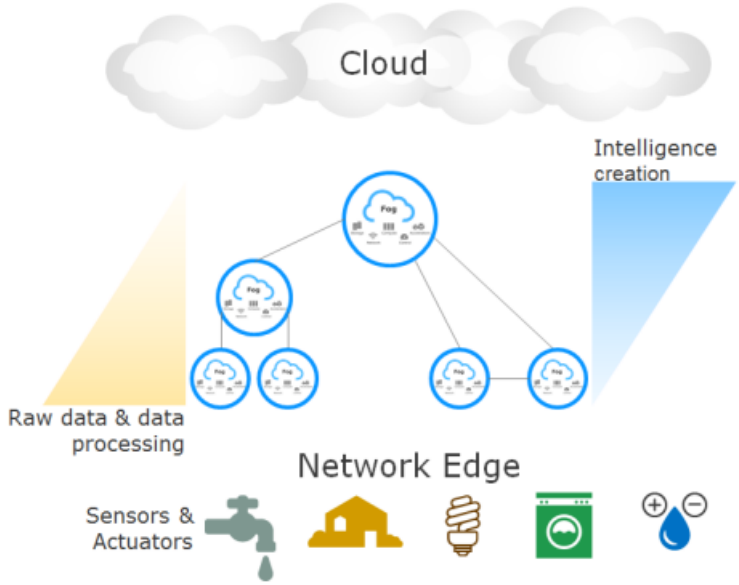

FIGURE 2.6 – Architecture à N niveaux du Fog computing [15].

2.4.3 Principales caractéristiques du Fog computing

1. **Distribution géographique étendue :** Le Fog peut couvrir de vastes zones, permettant aux applications d'être réparties sur différents régions.
2. **Adaptation à la mobilité et à la géolocalisation :** Le Fog s'ajuste dynamiquement aux déplacements des utilisateurs et des équipements, en tenant compte de leur localisation et de la portée des applications.
3. **Sécurité renforcée :** Le Fog améliore la sécurité en offrant des nœuds intermédiaires avec des protocoles de communication plus robustes. Il permet également de traiter les données au plus près des utilisateurs, limitant ainsi la transmission d'informations sensibles vers le Cloud.
4. **Autonomie et flexibilité :** Le Fog se caractérise par sa capacité à gérer de manière autonome le cycle de vie des applications. La virtualisation permet une programmabilité et une reconfiguration aisées, assurant l'adaptabilité de l'infrastructure dans un environnement dynamique [35].

2.5 Comparison entre les trois architectures

Le cloud computing, le fog computing et l'edge computing diffèrent principalement par l'endroit où les données sont traitées et stockées. Le cloud est un stockage centralisé distant des terminaux, idéal pour l'analyse de grandes quantités de données. Le fog est une couche intermédiaire qui combine les avantages du cloud et de l'edge, permettant une analyse en temps réel et une réponse rapide aux événements. Contrairement à l'edge, le fog a une architecture réseau. L'edge est le plus proche des périphériques terminaux, offrant la latence la plus faible et une réponse immédiate aux données. Il permet le calcul et le stockage de données directement sur les périphériques, les applications et les passerelles edge [3].

	Cloud	Fog	Edge
Latence	La plus élevée	Moyenne	La plus faible
Évolutivité	Élevée, facile à monter en échelle	Évolutive dans le réseau	Difficile à monter en échelle
Analyse des données	Traitement de données moins sensibles au temps, stockage permanent	Temps réel, décide de traiter localement ou d'envoyer vers le cloud	Prise de décision en temps réel, instantanée
Puissance de calcul	Élevée	Limitée	Limitée
Interopérabilité	Élevée	Élevée	Faible
Confidentialité et sécurité	Faible	Faible	Elevée
Niveau de traitement	Centrale	Serveur Fog	Serveur Edge
Efficacité	Faible	Elevée	Trés Elevée

Tableau 2.1 – Comparaison entre Cloud, Fog et Edge.

2.6 Machine Learning pour l'IoT

2.6.1 Définition

Le machine learning, connu sous le nom d'apprentissage automatique, est une discipline qui se concentre sur l'utilisation des ordinateurs pour imiter les processus d'apprentissage humain et développer des méthodes permettant aux machines de s'améliorer de manière autonome. Cela inclut l'acquisition de nouvelles compétences et connaissances, la reconnaissance des informations existantes et l'amélioration continue des performances. Comparé à l'apprentissage humain, l'apprentissage automatique est plus rapide, facilite l'accumulation des connaissances et permet une diffusion plus efficace des résultats. Ainsi, chaque progrès réalisé dans ce domaine renforce les capacités des ordinateurs, impactant ainsi la société de manière significative [21].

2.6.2 Techniques et Applications de Machine Learning dans l'IoT

2.6.2.1 Techniques

1. **Techniques de Machine Learning Supervisé :** Le machine learning supervisé repose sur l'utilisation de données étiquetées pour entraîner des modèles capables de prédire des résultats ou de classer des informations. Les algorithmes apprennent à partir d'un ensemble de données d'entraînement et sont ensuite utilisées pour faire des prédictions sur de nouvelles données.

    - **Modèles de Régression :** La régression est utilisée pour prédire une valeur continue. Par exemple, un modèle de régression peut être utilisé pour prévoir les prix de l'immobilier, les températures ou les ventes futures d'un produit.
    - **Modèles de Classification :** La classification a pour objectif de catégoriser des données en classes prédéfinies. Par exemple, un modèle de classification peut être utilisé

pour déterminer si un e-mail est un spam ou non, ou pour diagnostiquer une maladie en fonction des symptômes.

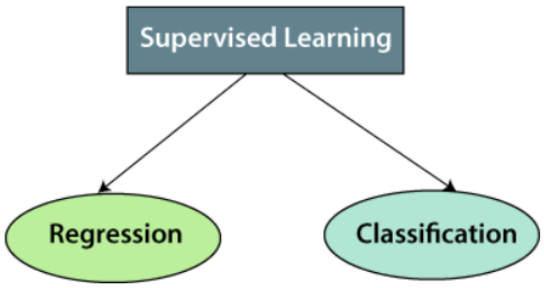

FIGURE 2.7 – Types de machine learning supervisé [18].

2. **Techniques de Machine Learning Non Supervisé :** Le machine learning non supervisé travaille avec des données non étiquetées pour découvrir des structures ou des motifs cachés. Les algorithmes tentent de regrouper les données ou de détecter des anomalies sans intervention humaine directe.

   - **Techniques de Clustering :** consiste à regrouper des données similaires en clusters ou groupes. Par exemple, un modèle de clustering peut être utilisé pour segmenter des clients en fonction de leurs comportements d'achat, ou pour identifier des groupes de patients présentant des symptômes similaires [18].
   - **Techniques d'association :** est une technique d'apprentissage utilisée pour découvrir les relations entre les variables dans de grandes bases de données. Elle permet d'identifier les ensembles d'éléments qui apparaissent fréquemment ensemble dans le jeu de données. Les règles d'association peuvent rendre les stratégies marketing plus efficaces. Par exemple, les personnes qui achètent un article X (comme du pain) ont également tendance à acheter un article Y (comme du beurre ou de la confiture). Un exemple typique de règle d'association est l'analyse du panier d'achat [9].

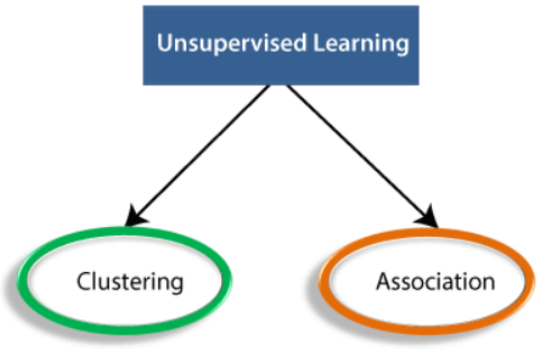

FIGURE 2.8 – Types de machine learning non supervisé [9].

3. **Techniques de Reinforcement Learning (Reinforcement Learning (RL)) :** Le RL est une technique d'apprentissage machine où un système apprend par essais et erreurs en interagissant avec un environnement. Le système effectue des actions et reçoit des récompenses ou des pénalités en fonction des résultats obtenus. L'objectif est de trouver une stratégie qui maximise la récompense totale à long terme. Dans les réseaux IoT, le RL peut être utile car il ne nécessite pas d'énormes données d'entraînement, mais présente des défis comme une lente convergence et la nécessité de connaître la dynamique de l'environnement.

4. **Techniques de Deep Learning (DL) :** Le DL est une branche du machine learning utilisant des réseaux de neurones profonds pour modéliser des relations complexes dans les données. Il est particulièrement performant pour traiter des données non structurées comme les images, textes et sons. Pour la sécurité IoT, le DL offre de bonnes performances avec une faible latence et complexité comparé aux techniques classiques. Les réseaux profonds peuvent aussi extraire des représentations de basse dimension à partir de données brutes de haute dimension [18].

2.6.2.2 Applications

Les recherches montrent que la technologie d'apprentissage automatique est largement utilisée dans les domaines du marketing, de la finance, des télécommunications et de l'analyse de réseaux. Dans le domaine du marketing, la technologie d'apprentissage automatique est plus fréquemment utilisée pour les tâches liées à la classification. En finance, elle est davantage appliquée aux tâches de prévision. Dans le domaine de l'analyse de réseaux, cette technologie est employée pour des tâches connexes. Dans le secteur des télécommunications, l'apprentissage automatique est largement utilisé pour des tâches de classification, de prédiction et de détection.
De plus, l'apprentissage automatique trouve également des applications dans le domaine de l'exploration de données, en combinaison avec d'autres applications. Les méthodes typiques comprennent l'initialisation basée sur les réseaux de neurones, l'application du calcul évolutionnaire dans la recherche en apprentissage automatique, l'étude de la classification par niveaux de l'apprentissage automatique, et l'apprentissage automatique basé sur les ensembles approximatifs, entre autres [21].

2.6.3 Modèles de Machine Learning

Il existe de nombreux modèles de machine learning, chacun adapté à des tâches spécifiques. Voici quelques-uns des modèles les plus couramment utilisés pour détection et la prédiction :

1. **Supervisé :**
    - **Régression linéaire :** La régression linéaire vise à trouver la ligne qui correspond le mieux aux données. Ses extensions incluent :
        - **Régression linéaire multiple :** Cherche le meilleur plan pour ajuster les données.
        - **Régression polynomiale :** Cherche la meilleure courbe pour ajuster les données.

- **Arbre de décision (Decision Tree) :** Les arbres de décision sont des modèles populaires utilisés en recherche opérationnelle, planification stratégique et apprentissage automatique. Ils sont représentés sous forme d'arborescence, où chaque rectangle est un noeud. Plus il y a de noeuds, plus l'arbre est généralement précis pour représenter les données. Les noeuds finaux où les décisions sont prises sont appelés les "feuilles". Bien qu'intuitifs et faciles à construire, les arbres de décision manquent parfois de précision.
- **Forêt d'arbres décisionnels (Random Forest) :** Une forêt aléatoire est un ensemble de plusieurs arbres de décision. Chaque arbre est construit à partir d'un échantillon aléatoire des données d'entraînement. De plus, pour chaque noeud de l'arbre, un sous-ensemble aléatoire de variables (features) est sélectionné parmi lesquelles choisir la meilleure division. Ainsi, chaque arbre individuel donne une prédiction. La prédiction finale de la forêt est déterminée par un vote majoritaire entre tous les arbres.
- **Les réseaux de neurones (neural networks) :** Un réseau de neurones est un modèle computationnel constitué de multiples couches interconnectées, s'inspirant de la structure et du fonctionnement du cerveau humain. Il comprend une couche d'entrée pour les données, une ou plusieurs couches cachées qui appliquent des fonctions mathématiques, et une couche de sortie qui fournit le résultat. Bien que basé sur des calculs complexes, il permet d'analyser des données de manière analogue, mais simplifiée, au cerveau humain.
- **Support Vector Machine (SVM) :** c'est un modèle qui cherche à tracer la meilleure ligne de séparation entre deux groupes de données. Cette ligne, appelée hyperplan, est choisie pour maximiser la distance avec les exemples les plus proches de chaque groupe. Cela permet d'obtenir

une frontière claire pour bien classer de nouvelles données dans l'un ou l'autre groupe. Bien que basé sur des lignes droites, le SVM peut aussi représenter des séparations plus complexes grâce à des calculs mathématiques.

- **K-Nearest Neighbor (K-NN) :** est un algorithme de classification utilisé en apprentissage automatique. Il fonctionne en classant un échantillon de données en fonction des classes des K échantillons les plus proches dans l'espace des caractéristiques. Plus précisément, pour classer un nouvel échantillon, l'algorithme identifie les K échantillons du jeu de données d'entraînement qui sont les plus proches de cet échantillon (selon une métrique de distance, généralement la distance euclidienne), et attribue à l'échantillon la classe majoritaire parmi ces K voisins [24].
- **AutoRegressive Integrated Moving Average (ARIMA) :** utilisé pour l'analyse et la prévision de séries temporelles. Il est particulièrement efficace pour modéliser des séries temporelles complexes et est largement utilisé dans divers domaines tels que l'économie, la finance et la météorologie pour effectuer des prévisions à court terme [2].
- **Random Forest Regression :** il repose sur l'utilisation d'un ensemble d'arbres de décision pour effectuer des tâches de régression. Dans ce processus, plusieurs arbres de décision sont créés en utilisant des sous-ensembles aléatoires des données d'entraînement. Chaque arbre génère des prédictions sur la variable cible, et la prédiction finale est obtenue en prenant la moyenne des prédictions de tous les arbres [10].

2. **Non Supervisé :**

- **Analyse en composantes principales (Principal Component Analysis (PCA)) :** modèle de réduction

de dimensionnalité non supervisé très utilisé en analyse d'un jeu de données, c'est-à-dire le nombre de variables qui le décrivent, tout en conservant un maximum d'informations. Elle projette les données initiales dans un nouvel espace de dimension inférieure, dont les nouvelles dimensions appelées composantes principales sont des combinaisons linéaires des dimensions d'origine choisies pour capturer l'essentiel de la variance des données [11].

- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN) :** est un algorithme de clustering qui utilise la densité spatiale des données pour former des groupes et identifier des anomalies. Il se distingue par sa capacité à détecter des clusters de formes variées en identifiant les zones de haute densité séparées par des zones de faible densité, sans présupposer de forme spécifique pour les clusters [30].
- **Isolation Forest :** est un modèle de détection d'anomalies, basé sur le principe que les anomalies sont "rares et différentes". Il construit une forêt d'arbres de décision en partitionnant aléatoirement l'espace des données. Les points nécessitant moins de partitions pour être isolés sont considérés comme des anomalies potentielles. Chaque point de données reçoit un score d'anomalie basé sur la profondeur moyenne des feuilles atteintes dans tous les arbres. Ce modèle est apprécié pour sa simplicité conceptuelle, son efficacité computationnelle et sa capacité à traiter de grands ensembles de données sans nécessiter de mesures de distance ou de densité [32].
- **One-Class SVM** utilisée pour la détection d'anomalies. Il permet de déterminer si un nouveau point de données est similaire ou différent des données sur lesquelles il a été entraîné .
- **Local Outlier Factor (LOF) :** efficace pour détecter les anomalies dans les données en identifiant les points

qui ont une densité locale significativement différente de celle de leurs voisins. En évaluant cette densité locale, le LOF permet de repérer les points qui se démarquent de leur environnement en termes de densité, facilitant ainsi la détection d'anomalies [4].

2.6.4 Détection des Anomalies, Clustering et Prédiction

2.6.4.1 Détection des Anomalies

1. **Définition de la détection des anomalies :** La détection des anomalies est le processus d'identification de points de données qui s'écartent de manière significative de la norme ou des comportements attendus dans un ensemble de données ou un système.

2. **Techniques de détection des anomalies :** se divisent principalement en deux catégories : celles basées sur des règles et celles utilisant le (Machine Learning (ML).

   - **Basées sur des règles :** Une liste de conditions décrit ce qui est considéré comme "normal"
   - **Catégories de techniques** ML :
     - **Supervisées :** Utilisent des données étiquetées pour apprendre à distinguer le normal de l'anormal.
     - **Non supervisées :** Fonctionnent sans étiquettes, en identifiant les schémas rares ou divergents.
     - **Semi-supervisées :** Combinent données étiquetées et non étiquetées pour établir une référence et détecter les écarts.

Le choix de la méthode dépend du contexte, des données disponibles et des objectifs spécifiques de la détection d'anomalies [7].

2.6.4.2 Clustering

Le clustering est une technique d'apprentissage non supervisé qui consiste à regrouper des données similaires en différents groupes ou clusters. Il existe plusieurs méthodes de clustering, parmi lesquelles :

1. **Les méthodes hiérarchiques :** Elles forment des connexions pas à pas entre les individus proches, généralement représentées par un dendrogramme. Elles nécessitent de choisir une mesure de distance entre individus et une stratégie d'agrégation (lien simple, complet, moyenne, etc).
2. **Les méthodes centroïdes :** La plus connue est la méthode des k-moyennes. Elle initialise k centres de classes au hasard, associe chaque individu au centre le plus proche, calcule les nouvelles moyennes des classes, et réitère jusqu'à convergence. Le nombre de classes k doit être choisi à l'avance.
3. **Les méthodes à densité :** Elles identifient les zones de forte densité comme des clusters, et considèrent les points isolés comme du bruit. L'algorithme DBSCAN utilise les paramètres de distance de voisinage et de taille minimale de voisinage N [8].

2.6.4.3 Prédiction

La prédiction en machine learning est le processus par lequel un modèle, entraîné sur des données historiques et actuelles, estime ou infère un résultat futur, une classe ou une valeur numérique pour de nouvelles données [5].

Dans la partie suivante, nous synthétiserons les travaux connexes qui ont exploité ces architectures et ces modèles techniques de machine learning dans le contexte de l'IoT

Deuxième partie

Partie II : Revue des travaux connexes

2.7 Synthèse des recherches et projets existants

Dans cette partie, nous étudierons les différentes solutions proposées dans la littérature, notamment pour les réseaux de véhicules autonomes, l'e-Health, les maisons intelligentes, l'industrie et d'autres applications critiques, afin de répondre aux défis de communication et de traitement de données dans des environnements nécessitant une faible latence et une haute fiabilité :

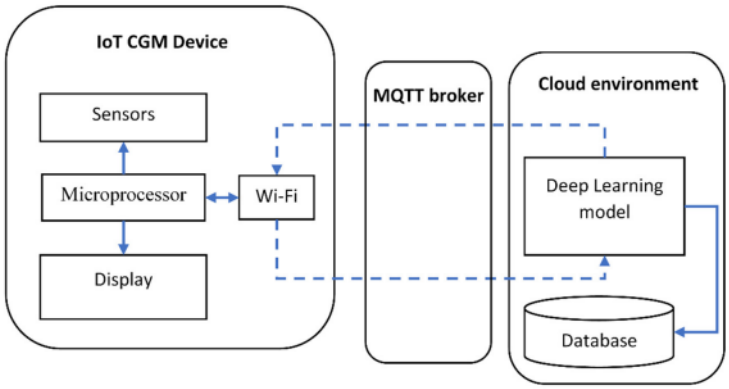

FIGURE 2.9 – Architecture du système proposée [13].

La Figure 2.9 illustre une architecture basée sur le cloud pour améliorer la prédiction des niveaux de glucose sanguin pour les patients atteints de diabète de type 1. Dans cette figure, on peut voir que le système comprend trois composantes principales : un dispositif IoT portable avec capteur de glucose et affichage, un courtier MQTT assurant la connexion entre le dispositif IoT et l'environnement cloud, et un modèle de prédiction basé sur le Deep Learning déployé dans le cloud.
Le dispositif IoT portable intègre un capteur mesurant en continu les niveaux de glucose dans le sang du patient. Ces données sont transmises via Wi-Fi au courtier MQTT, qui les transmet au cloud.ces données sont stockées dans une base de données. Un ensemble de 20 points de données des 100 dernières minutes est alors utilisé pour alimenter le modèle Deep Learning de type Recurrent Neural Network-Restricted Boltzmann Machine (RNN-RBM). Ce modèle analyse les ten-

dances temporelles pour prédire les futurs niveaux de glucose du patient jusqu'à 30 minutes à l'avance. Les prédictions générées par le modèle RNN-RBM déployé dans le cloud sont ensuite renvoyées au dispositif IoT portable du patient pour être affichées, lui permettant d'anticiper les fluctuations de sa glycémie.

Les auteurs de [13] proposent une architecture basée sur le cloud computing pour améliorer la prédiction des niveaux de glucose sanguin pour les patients atteints de diabète de type 1. Leur approche a été entraîné sur un ensemble de données réelles provenant du Diabetes Research in Children Network (DirecNet), constitué de mesures de glycémie pour 110 patients âgés de 7 à 17 ans, atteints de diabète de type 1. Pour l'évaluation, ils ont sélectionné aléatoirement un sous-ensemble de 10 patients parmi ceux ayant plus de 1000 points de données de niveaux de glycémie chacun. Pour chacun de ces 10 patients sélectionnés, 80% de leurs données individuelles de niveaux de glycémie ont été utilisées pour entraîner le modèle, tandis que les 20% restants ont servi aux tests d'évaluation. La performance a été mesurée en termes d'erreur quadratique moyenne (Root Mean Squared Error (RMSE)). Leur approche combinant le cloud et le deep learning a permis d'atteindre une erreur RMSE moyenne de 15,589 sur les 10 patients testés, surpassant ainsi les méthodes existantes de prédiction de la glycémie dans la littérature.

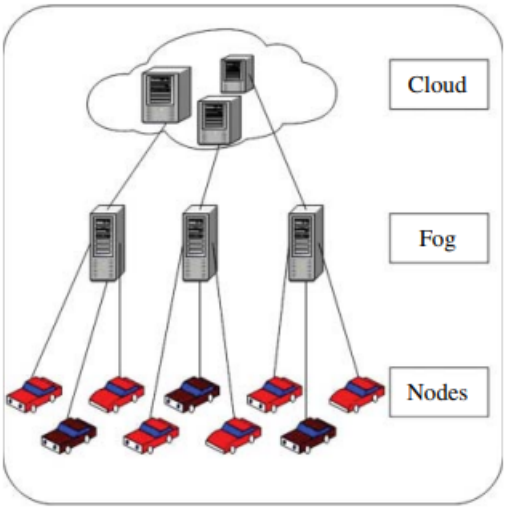

FIGURE 2.10 – Architecture basée sur le Fog proposée pour la communication entre les véhicules autonomes [34].

La Figure 2.10 illustre une architecture basée sur le fog proposée pour la communication entre les véhicules autonomes. Dans cette figure, on peut voir que tous les nœuds, qui sont des véhicules autonomes, communiquent avec le fog au lieu de communiquer directement avec le cloud. Certaines décisions et traitements nécessitant une faible latence sont effectués au niveau des noeuds fog. Si ces noeuds ne peuvent pas traiter les données, elles sont alors transmises vers le cloud pour un traitement supplémentaire.

Les auteurs de [34] ont proposé une architecture a trois couches (Couche véhicules autonomes, couche Fog et couche Cloud) basée sur le Fog pour accroître la fiabilité des Réseaux véhiculaires autonomes (AVN). La solution proposée utilise des nœuds de fog pour permettre un traitement des données à Proximité du réseau, réduisant ainsi la latence et la bande passante nécessaires par rapport à une architecture cloud traditionnelle. d'après leurs évaluations de la latence et l'utilisation du réseau avec 10 puis 100 véhicules on note :

Latence :

- Avec 10 véhicules, la latence passe de 7,03 ms (solution cloud) à 0,457 ms (solution proposée), ce qui représente une amélioration significative.

- Avec 100 véhicules, la latence passe de 4509,8 ms (solution cloud) à 3,671 ms (solution proposée), ce qui représente également une amélioration substantielle.

Utilisation réseau :

- Avec 10 véhicules, l'utilisation réseau passe de 22096,8 (solution cloud) à 509,49 (solution proposée), ce qui représente une réduction importante de la bande passante utilisée.
- Avec 100 véhicules, l'utilisation réseau passe de 129553,84 (solution cloud) à 5094,9 (solution proposée), ce qui représente également une réduction considérable de la bande passante utilisée.

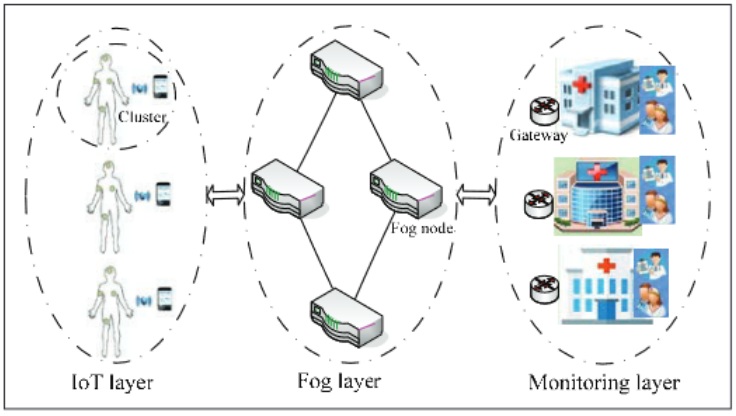

FIGURE 2.11 – Architecture Fog pour traitement des données des patients [36].

La figure 2.11 illustre l'architecture à trois couches du système de soins de santé assisté par le fog computing proposé. La couche IoT comprend les capteurs déployés sur le patient et un appareil intelligent centralisateur. La couche fog contient les nœuds fog qui traitent et mettent en cache les données du patient provenant de la couche IoT. La couche monitoring au niveau de l'hôpital permet aux nœuds de récupérer les données du patient auprès du fournisseur le plus proche ou du nœud fog le plus proche via un système de nommage.

Les auteurs dans [36] ont traité un système IoT de traitement de données des patients qui avaient des limitations dans

ces capacités de traitement et du stockage dans ces noeuds IoT et par conséquence la haute latence.La solution qu'ils ont proposée consiste a utiliser le Fog computing pour surmonter les limitations de ressources des nœuds de l'IoT, d'aprés leur évaluation avec 10 noeuds fog,10 clusters,50 noeuds IoT et 10 passerelles sur ns3,les résultats montrent qu'avec le fog computing on peut réduire la latence pour récupérer les données des patients par 28.5 % qui assure la rapidité et l'efficacité par rapport a l'utilisation des noeuds IoT Uniquement.

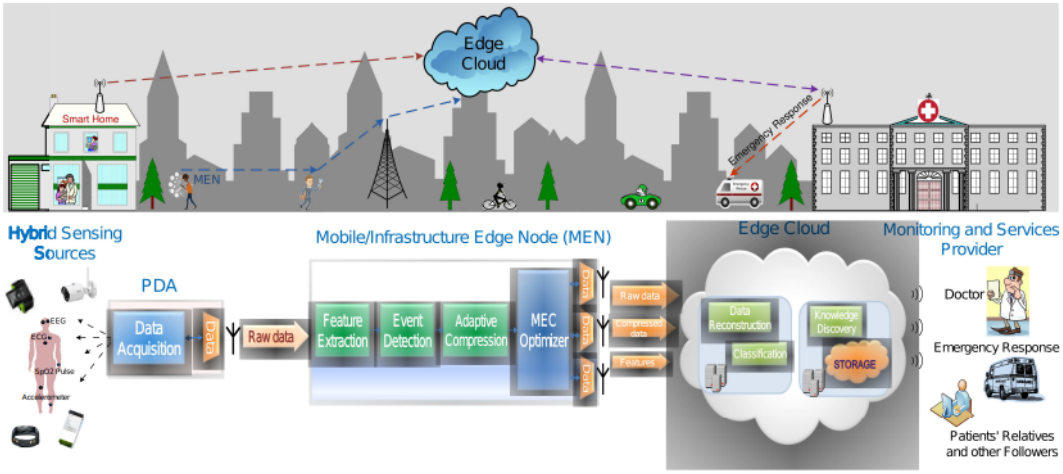

FIGURE 2.12 – Architecture basée sur le MEC pour un système de santé intelligent [12].

La figure 2.12 illustre l'architecture proposé pour système de santé intelligente (s-health) basée sur l'intégration du MEC. Les données sont d'abord générées par diverses sources auprès du patient(les réseaux de capteurs corporels, les caméras IP, les smartphones et les appareils médicaux externes). Ces données sont ensuite agrégées par un agrégateur de données patient Patient Data Aggregator (PDA) qui fait le role d'un hub de communication. Le PDA transmet alors les données collectées depuis les patients au nœud de edge mobile/infrastructure MEN. C'est à ce niveau que s'effectue le traitement clé des données :
- Il applique un traitement sur les données brutes, comme la compression, l'extraction de caractéristiques, la classification
- Il détecte les situations d'urgence basées sur l'analyse des données Après ce traitement au niveau du MEN, seules les données

compressées, les caractéristiques extraites ou les alertes d'urgence sont transmises au edge cloud pour un stockage et une analyse plus poussée. Le edge cloud, qui peut être un hôpital par exemple, effectue alors le stockage des données, les analyses avancées pour la détection de tendances, la gestion de la santé de la population, etc.

Les auteurs dans [12] ont étudier plusieurs application de s-health déjà existantes (détection des troubles cardiaques, surveillance cardiaque à distance, mesure de la fréquence cardiaque sans contact, etc.) qui implémente le cloud computing uniquement,cette implémentation souffre de plusieurs problèmes nottament la haute latence car certaines applications nécessitent des réponses en temps réel comme la détection des troubles cardiaques,aussi le problème de la congestion du réseau a cause des grandes quantités de données qui sont envoyés vers le cloud.Alors les auteurs ont proposé une architecture de système de santé intelligente (s-health) basée sur l'intégration du Multi-access Edge Computing (MEC) dans ces applications au lieu du cloud computing unique , cette solution permet d'exploiter la possibilité de faire le traitement et le stockage des données à proximité du réseau (selon la priorité du traitement et la capacité de traitement nécessaire) au lieu d'envoyer tous les données vers le cloud,afin de répondre aux exigences de ces applications. D'aprés leurs tests, ils ont trouvé ces résultats :

- Pour les applications de surveillance continue générant de gros volumes de données (surveillance cardiaque, détection d'ischémie, Parkinson, rythme cardiaque), le MEC permet de réduire les volumes de données transmise grâce à la technique de compression faite au niveau Edge, économisant aussi la bande passante par rapport au cloud.

- Pour les applications nécessitant une détection/prédiction rapide d'événements critiques (bradycardie, crises d'épilepsie, changements Électrocardiogramme (ECG), bronchopneumo-

pathie), le MEC offre l'avantage d'une faible latence car les tâches sont faites prés des patients (source des données).

- l'utilisation du MEC permet une réduction de la consommation d'énergie, ce qui constitue un avantage majeur en prolongeant l'autonomie des appareils IoT comme dans le cas de la surveillance cardiaque à distance.

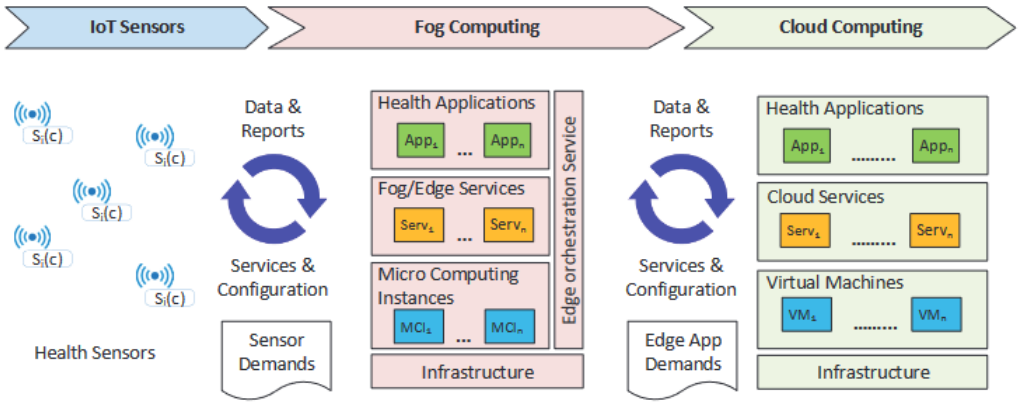

FIGURE 2.13 – Architecture Fog pour traitement des données des patients [31].

La figure 2.13 illustre l'architecture hybride (Fog/Cloud) proposé pour les applications healthcare etudiées.

les données de santé sont collectées à partir de divers capteurs de santé, ensuite transmises au niveau du fog computing. Le traitement des données de santé IoT est distribué entre le niveau Fog et le niveau Cloud. Au niveau Fog, les données brutes provenant des capteurs sont pré-traitées par les applications et services de Fog s'exécutant sur les instances de micro-calcul. Ensuite, les données pré-traitées et les rapports sont transmis au Cloud pour subir des traitements supplémentaires par les applications et services Cloud s'exécutant sur les machines virtuelles. En ce qui concerne le stockage, les données brutes et/ou pré-traitées peuvent être temporairement stockées au niveau de l'infrastructure Fog, tandis que les données finalement traitées, les rapports et éventuellement les données brutes sont stockés de manière persistante au sein de l'infrastructure Cloud centralisée.

Dans l'article [31] les auteurs proposent une solution basée sur le fog/cloud computing hybride dans des applications de santé. Ils comparent ses performances par rapport à une architecture basée uniquement sur le cloud computing. L'approche proposée consiste à utiliser des Micro Computing Instance (MCI) qui sont des petites instances de calcul virtualisées avec des ressources limitées (CPU, mémoire, stockage), déployée sur un nœud Fog pour exécuter un module d'application.Cette dernière coute moins en termes d'énergie et d'argent,on trouve aussi qu'un ensemble de MCI exécute une seule application contrairement au cloud pure qui utilise une instance Virtual Machine (VM) pour exécuter une application,le VM est couteux en termes d'énergie et d'argent.Aussi,les taches qui sont urgents et/ou ne nécessitant pas des grandes capacités de traitements seront traités au niveau Fog,par contre le cloud est utilisé lorsque ces taches dépassent la capacité du fog.Les auteurs ont utilisé l'outil iFogSim pour leurs tests avec ces paramètres :

- **Architecture cloud pure :** latence a partir du source est 100ms, le cout de VM est 0.8-0.12 usd/min,la consommation de l'énergie par VM est 10-15 MegaJoules,le nombre moyen des VMs par serveur est 10-15 VMs.
- **Architecture Fog/cloud hybride proposée :**latence a partir du source est 10ms, le cout de MCI est 0.01-0.03 usd/min,la consommation de l'énergie par MCI est 2-3 MegaJoules,le nombre moyen des MCIs par serveur est 3-10 VMs.

D'aprés leurs tests ils ont conclu que leur solution proposée :

- améliore les performances en termes de la distribution du service et le cout des instances par rapport au cloud.
- Réduisent la latence car avec 30 applications healthcare en exécution elle passe d'une moyenne de 150ms (solution cloud pure) vers les environs de 80ms avec cette solution.

- Réduisent aussi la consommation de l'energie passant dans le cas de 30 applications healthcare utilisées de 375 Mega Joules (solution cloud) vers moins de 275 MJ dans cette solution.

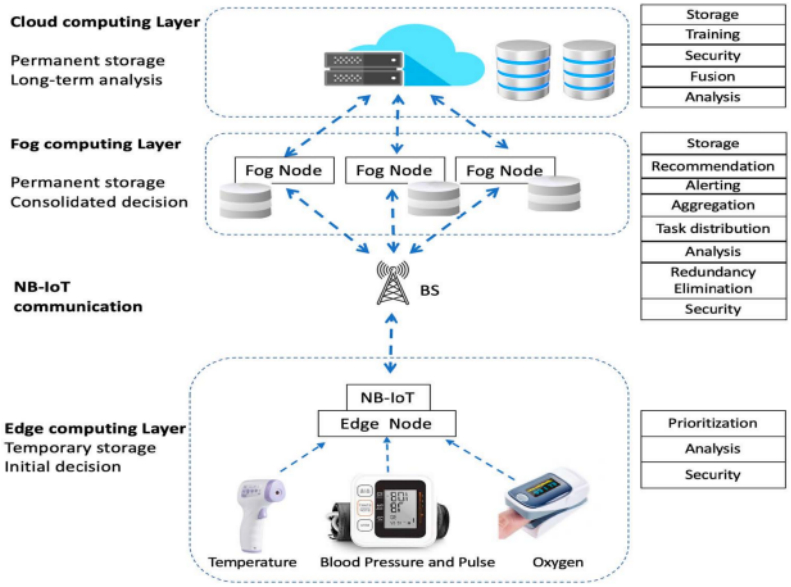

FIGURE 2.14 – L'architecture hiérarchique du système composée d'une couche edge computing, couche fog computing et couche cloud computing [37].

La figure 2.14 illustre l'architecture du système composée d'une couche edge computing, couche fog computing et couche cloud computing pour la surveillance des signes vitaux des patients dans les zones rurales.

- La couche edge est constituée de nœuds edge connectés aux appareils médicaux comme les capteurs de température, de pression artérielle, de pouls et d'oxygène. À ce niveau, les données sont analysées, classifiées en données normales ou anormales à l'aide d'un algorithme d'arbre de décision. Cette classification initiale permet de prendre des décisions préliminaires sur l'état de santé du patient. Ensuite, la couche Edge priorise la transmission des données anormales avant les données normales afin de réduire l'engorgement au niveau des stations de base NB-IoT, diminuant ainsi les délais de communication. Les données normales sont stockées temporairement à la couche Edge et transmises quand le canal NB-IoT est moins encombré.

- La couche Fog reçoit les données pré-traitées de la couche Edge et effectue une analyse approfondie à l'aide d'algorithmes machine learning. Elle vise à réduire la charge de calcul sur le cloud en fournissant un environnement virtualisé pour le calcul et le stockage de données. La couche Fog agrège les données médicales des patients, génère des diagnostics détaillés grâce à des ressources de calcul supérieures, et notifie les prestataires avec des recommandations en cas de problème détecté.

- La couche Cloud gère l'ensemble du système de monitoring de santé. Elle reçoit les analyses de la couche Fog pour stockage permanent. Grâce à sa puissance de calcul supérieure, la couche Cloud effectue la fusion des données agrégées, les analyses avancées à long terme, les prédictions long terme, et fournit des recommandations aux autorités de santé. Son rôle est d'assurer le traitement final des données.

Les auteurs de [37] proposent une architecture hybride (Edge/Fog/Cloud) pour la surveillance des signes vitaux des patients dans les zones rurales où les ressources médicales sont très limitées. La nouveauté réside dans l'utilisation de la technologie NarrowBand Internet des Objets (NB-IoT). Il utilise NB-IoT au lieu des anciennes technologies pour couvrir un très grand nombre de dispositifs médicaux sur de larges zones géographiques tout en réduisant considérablement la consommation d'énergie de ces dispositifs.
Selon leur évaluation à l'aide des outils de simulation CloudSim, iFogSim et ns3-NB-IoT sur des données réelles de signes vitaux médicaux, les résultats obtenus sont les suivants :

- **Résultats de délai NB-IoT moyen (TR) :** L'architecture proposée réduit le délai NB-IoT de 59,9% par rapport aux autres configurations évaluées pour un grand nombre de dispositifs (200 dispositifs). Cette réduction est attribuée à la couche Edge qui priorise la transmission des données anor-

males afin de minimiser l'engorgement au niveau des stations de base NB-IoT.

- **Résultats de temps d'exécution (TC) :** L'architecture proposée réduit le temps d'exécution de 52,6% par rapport à la configuration sans Edge ni Fog, de 20,5% par rapport à la configuration sans Edge et de 42,3% par rapport à la configuration sans Fog. En moyenne, la réduction du temps d'exécution est de 38,5% grâce à la répartition des tâches entre les couches Edge, Fog et Cloud.
- **Résultats de temps d'authentification (TA) :** Parmi les protocoles d'authentification évalués le protocole Light-Edge présente le meilleur temps d'authentification. Pour 200 dispositifs, Light-Edge permet une réduction de 35,1% du temps d'authentification par rapport aux autres protocoles.

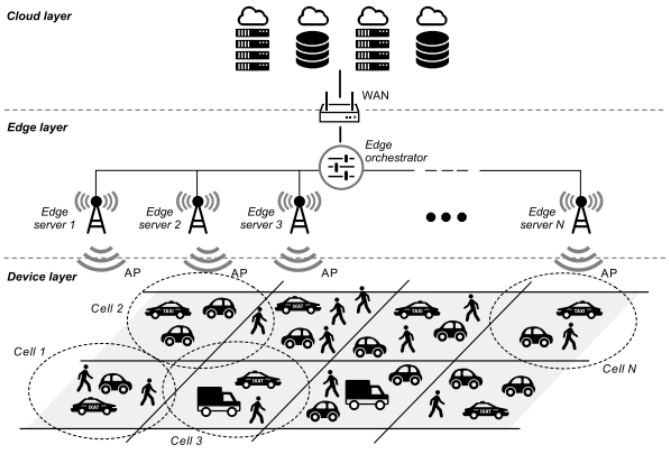

FIGURE 2.15 – architecture hybride (Edge/cloud) proposée [26].

La figure 2.15 illustre l'architecture Hybride (Edge/Cloud) proposé pour gérer efficacement les tâches liées à la mobilité urbaine : La couche des appareils comprend les objets IoT (taxis, voitures, smartphones) qui génèrent des données géolocalisées. La couche edge comprend des serveurs répartis géographiquement qui collectent et pré-traitent les données brutes générées. La couche cloud fournit des ressources de calcul et de stockage massives pour exécuter les tâches les plus lourdes.

Les auteurs [26] proposent une architecture hybride edge/cloud pour des applications de prédiction et la planification de la mobilité urbaine (Prédiction de la prochaine localisation pour les taxis, Publicité basée sur la localisation pour les voitures, Recommandation de points d'intérêt pour les touristes.) et comparent ses performances par rapport à deux architectures basée uniquement sur le cloud computing et Edge computing. Un composant clé appelé Edge Orchestrator décide, pour chaque tâche entrante, si elle doit être exécutée sur l'infrastructure edge locale ou envoyée vers le cloud. Il utilise deux politiques d'orchestration à cet effet :

- La politique basée sur l'utilisation du réseau (edge/cloud-NB) mesure le délai réseau de l'appareil IoT vers le cloud pour décider où exécuter la tâche. Elle envoie une tâche factice de 1 Mo pour mesurer le délai d'aller-retour, qui inclut le délai de transmission des données et le délai de traitement par le cloud. Si l'utilisation de la bande passante dépasse un certain seuil (80% dans les expériences), la tâche est exécutée sur l'infrastructure edge locale pour éviter la congestion réseau. Sinon, elle est envoyée vers le cloud.
- La politique basée sur l'utilisation des ressources (edge/cloud-UB) surveille l'utilisation moyenne des ressources de calcul des serveurs edge. Si cette utilisation dépasse un certain seuil (80%), cela signifie que les ressources edge sont sous pression et les nouvelles tâches sont alors déportées vers le cloud. Sinon, elles sont gardées sur l'infrastructure edge.

Après simulations avec EdgeCloudSim, les résultats ont montré une réduction significative du temps de traitement (jusqu'à 87%) par rapport à l'architecture edge seule, une baisse drastique du taux d'échec des tâches (jusqu'à 40%) par rapport au cloud et edge seuls, ainsi qu'une meilleure utilisation des ressources (jusqu'à 38% de moins) que le cloud et l'edge seuls.

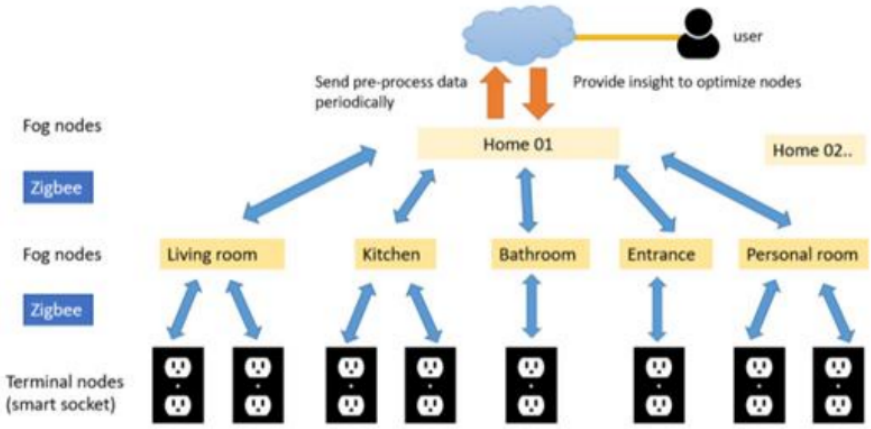

FIGURE 2.16 – architecture hybride (Fog/cloud) proposée [38].

La figure 2.16 illustre l'architecture Hybride (Fog/Cloud) proposé pour la gestion de la consommation énergétique dans les maisons intelligentes. L'architecture comporte trois couches principales : la couche des nœuds terminaux (prises intelligentes), la couche intermédiaire de fog nodes représentant des zones/salles spécifiques (salon, cuisine, salle de bain, entrée, chambre personnelle), et la couche supérieure de cloud. Les nœuds terminaux récupèrent les données de consommation électrique et les transmettent à la couche intermédiaire de fog nodes via ZigBee. La couche de fog nodes intermédiaire gère les nœuds terminaux d'une même zone/salle de la maison et peut réagir rapidement. La couche supérieure du cloud collecte les données agrégées de toutes les zones/salles de la maison à partir des fog nodes et communique périodiquement avec eux pour envoyer les insights d'optimisation.

Les auteurs [38] proposent une architecture hybride fog/cloud à trois couches pour la gestion de la consommation énergétique dans les maisons intelligentes, basée sur le protocole ZigBee. Cette architecture vise à résoudre les problèmes des architectures cloud seulement comme les pannes de serveur et la latence élevée. Ils ont évalué les performances en établissant deux réseaux différents sur le même dispositif : un réseau cloud seulement et

un réseau Hybride(Fog/cloud). Les métriques suivantes ont été mesurées :

- **Latence :** Mesurée en envoyant des paquets toutes les 3 secondes et en calculant le temps de traitement. L'architecture Hybride (Fog/cloud) a une latence nettement réduite de quelques millisecondes, sauf lors des mises à jour périodiques par le cloud. Le cloud-only avait une latence plus élevée, autour de 700ms à 2s.
- **Charge système moyenne :** Testée sur 15 minutes. Le Hybride(Fog/cloud) avec mises à jour périodiques du cloud avait une charge système plus faible que le cloud seulement.
- **Utilisation du CPU :** Testée sur 15 minutes avec des données provenant d'une, deux ou trois pièces. Le Hybride (Fog/cloud) avait une meilleure utilisation du CPU quand les données provenaient de plusieurs pièces, malgré des pics d'utilisation lors des mises à jour périodiques.

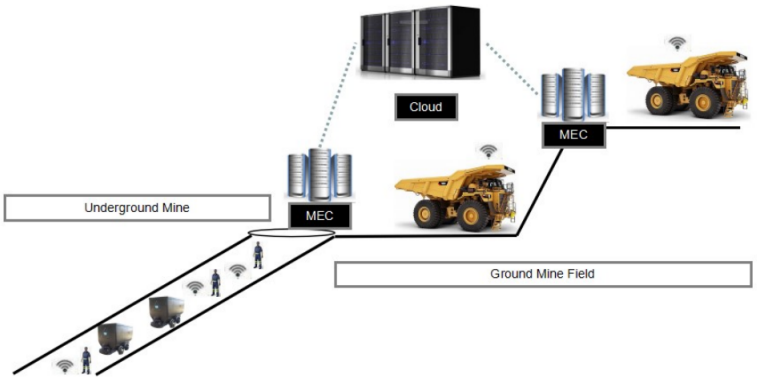

FIGURE 2.17 – Architecture MI-MEC proposée [27].

La figure 2.17 illustre une architecture hybride (Cloud/MEC) proposée pour la gestion des opérations minières. L'architecture comprend trois couches principales : la couche des équipements sur site (camions miniers), la couche intermédiaire des nœuds Edge (MEC) représentant les sites miniers spécifiques (mine souterraine, champ minier en surface), et la couche supérieure du

cloud. Les équipements sur site comme les camions miniers recueillent les données d'exploitation et les transmettent à la couche intermédiaire des nœuds MEC via un réseau sans fil. La couche intermédiaire des nœuds MEC gère les équipements d'un même site minier et peut réagir rapidement aux événements locaux. La couche supérieure du cloud collecte les données agrégées de tous les sites miniers à partir des nœuds MEC et communique périodiquement avec eux pour envoyer des analyses et optimisations. Les travailleurs ou capteurs déployés dans la mine souterraine sont également connectés aux nœuds MEC de cette couche intermédiaire pour permettre un monitoring et un contrôle des opérations.

Les auteurs [27] proposent une architecture appelée MI-MEC (Mobile Edge Computing pour l'industrie minière) conçue pour améliorer la sécurité dans les mines. La solution proposée consiste à utiliser le MEC pour traiter et analyser localement les données collectées par des capteurs IoT dans la mine, plutôt que d'envoyer ces données vers un cloud distant. Un cas d'usage spécifique détaillé est la prévention des explosions de gaz dans les mines de charbon souterraines. Pour cela, un modèle de machine learning est d'abord entraîné sur des données historiques afin de détecter les risques d'explosion à partir des niveaux de méthane et de poussières de charbon mesurés. Ensuite, si un tel risque est détecté par le modèle, un signal est immédiatement envoyé pour déclencher des volets coupe-feu et des alarmes afin d'alerter les mineurs.
Ils ont évalué les performances de leur approche à l'aide du simulateur EdgeCloudSim. Ils ont conduit des expériences pour tester les performances de leur proposition avec différents nombres de nœuds : 15 nœuds (5 camions et 10 mineurs), 30 nœuds (10 camions et 20 mineurs), 45 nœuds (15 camions et 30 mineurs) et 60 nœuds (20 camions et 40 mineurs) distribués sur une zone géographique de 1000m x 1000m, où les camions se déplacent à 15 m/s et les mineurs à 1 m/s. Cela permet d'examiner l'efficacité,

la faisabilité et l'évolutivité de leur proposition. Les résultats expérimentaux montrent que le temps de service et de traitement pour leur système de sécurité reste inférieur à 4 secondes pour analyser et renvoyer les résultats aux clients, ce qui est suffisamment rapide pour permettre une réaction à temps. De plus, le pourcentage très faible de tâches échouées traitées par le serveur edge garantit également la faisabilité de leur schéma.

Dans le tableau suivant, nous résumons les principales caractéristiques des différentes solutions étudiées, mettant en évidence leurs forces et faiblesses en termes de latence, bande passante, consommation d'énergie, sécurité et coûts.

				Evaluations					
Article	Application	Infrastructure	Outil utilisé	Latence	Bande passante	Consommation energie	Securité	Coûts	Congestion
[13]	prédiction des niveaux de glucose sanguin 1	Cloud	Deep Learning	/	/	Faible	/	/	/
[34]	Réseaux de véhicules autonomes	hybride (Fog/Cloud)	Simulation (iFogSim)	Réduite	Réduite	/	/	/	Réduite
[36]	Traitement de données des patients	Fog	NS3	Réduite	Réduite	Réduite	/	/	Réduite
[31]	Solutions healthcare	hybride (Fog/Cloud)	Simulateur iFogSim	Réduite	Réduite	Réduite	/	/	Réduite
[12]	Différentes applications dans smart health	Edge (MEC)	deep learning	Réduite	Réduite	Réduite	Réduite	coûts opérationnel réduite	Faible
[37]	Surveillance de la santé à distance	hybride(Fog/ Edge/Cloud)	Simulation (CloudSim, iFogSim et ns3-NB-IoT)	Réduite	Réduite	Efficace	Elevée (protocol lightweight pour authentification)	Coût d'installation augmenté a cause de la complexité	Réduite
[26]	Prédiction et la planification de la mobilité urbaine	hybride(Edge cloud)	Simulation (EdgeCloudSim)	Réduite	Réduite	/	/	/	/
[38]	gestion de la consommation énergétique dans les maisons intelligentes	Hybride (Fog/Cloud)	/	Réduite	/	/	/	/	/
[27]	Amélioration de la sécurité dans les mines	Hybride (Cloud/Edge (MEC))	EdgeCloudSim	Réduite	/	/	Réduite	/	/

Tableau 2.2 – Tableau récapitulatif des travaux de recherche étudiés.

Le tableau résume les résultats d'évaluation des performances de diverses solutions proposées pour des applications nécessitant un traitement des données massives. Ces solutions utilisent des architectures basées uniquement sur l'edge computing ou le fog computing, ou des solutions utilisant des architectures hybrides combinant le cloud computing avec le fog computing et/ou l'edge computing :

En ce qui concerne la latence, on remarque que tout les solutions proposées [34], [36], [31],[12], [37], [27], [26] et [38] parviennent à réduire la latence par rapport à une architecture cloud traditionnelle. Cela s'explique par le fait que le fog et l'edge computing permettent de traiter les données à proximité du réseau, évitant ainsi les allers-retours avec le cloud distant.
Pour la bande passante, les solutions [34], [36] , [31] , [12] ,[37] et [26] montrent une réduction de l'utilisation de la bande passante réseau. Cette économie est grace au pré-traitement,traitement et le filtrage des données fait a proximité soit au niveau edge ou fog, réduisant ainsi le volume de données transmis vers le cloud.
En ce qui concerne la consommation d'énergie, les solutions [36], [31] et [12] parviennent à la réduire. Cela est rendu possible grâce au traitement des données au niveau de l'edge computing (à la source même) et du fog computing (à proximité du réseau), évitant ainsi des transmissions sur de longues distances vers le cloud qui consomment plus d'énergie.

En ce qui concerne la sécurité, seules quelques solutions abordent cet aspect. La solution [37] propose un protocole léger d'authentification ("Light-Edge") qui permet d'améliorer significativement le temps d'authentification, renforçant ainsi la sécurité. Cependant, la plupart des autres travaux n'abordent pas ou peu les questions de sécurité des données. Du point de vue des coûts, on peut distinguer deux aspects : les coûts opérationnels et les coûts d'infrastructure initiaux. La solution [12] basée sur l'edge computing permet de réduire les coûts opérationnels par

rapport au cloud. En revanche, la solution hybride [37] voit ses coûts d'installation augmenter en raison de la complexité accrue de l'architecture à 3 niveaux (edge/fog/cloud). La plupart des autres travaux ne fournissent pas d'analyse détaillée des coûts.

Plusieurs solutions proposent des mécanismes pour réduire la congestion réseau vers le cloud. Dans [37], les données anormales sont envoyées en priorité depuis la couche edge pour limiter l'engorgement. Dans [34] et [36], une partie du traitement est effectuée au niveau du fog computing plutôt que dans le cloud. La solution [26] surveille l'utilisation du réseau et exécute les tâches localement sur l'edge computing si la bande passante est saturée. Ces approches visent à mieux répartir les charges et éviter la congestion sur les liens réseau montants.

2.8 Analyse critique des approches utilisées

D'après toutes les recherches que nous avons menées, il existe de nombreux travaux portant sur les trois architectures dans divers domaines, notamment dans le secteur médical et véhiculaire . Cependant, le domaine du commerce reste relativement peu exploré en comparaison, laissant ainsi un vide dans la littérature scientifique en ce qui concerne l'utilisation de ces architectures dans un contexte commercial. De plus, nous avons identifié plusieurs autres lacunes dans les recherches existantes :

- **Sécurité, confidentialité, respect de la vie privée :** Les articles [34], [13] , [12], [37], [38] et [26] n'abordent pas les mécanismes de protection des données sensibles (véhicules autonomes, santé), malgré que ce sont des aspects importants dans le choix d'une solution par rapport à une autre.
- **Gestion des ressources et la scalabilité :** Les articles [34], [13] , [36],[12] et [31] n'abordent pas suffisamment les défis liés à la gestion et à l'orchestration des ressources (calcul, stockage, bande passante) pour les nœuds fog/edge dis-

tribués, notamment pour la gestion dynamique et en temps réel des ressources pour les véhicules en mouvement [34].

- **Fiabilité et tolérance aux pannes :** L'article [36] et [27] omettent d'aborder la fiabilité et la tolérance aux pannes en cas de défaillance de nœuds.

2.9 Conclusion

Dans ce chapitre, nous avons étudié en détail les différentes architectures cloud, fog et edge computing, ainsi que les travaux existants proposant des solutions hybrides combinant ces paradigmes. Nous avons pu constater que chaque approche présente des forces et des faiblesses, et que le choix dépend des exigences spécifiques de l'application visée. Cependant, de nombreux défis restent à relever pour une adoption généralisée de ces architectures.

Les recherches faites dans le domaine du télécoms (aspect commerciale) sont limitées malgré l'importance de ce domaine, c'est pour cela que nous allons proposer dans le prochain chapitre la conception d'une architecture cloud hybride (Fog/Edge-Cloud) conçue spécialement pour satisfaire à ces besoins.

Chapitre 3

Conception, Implémentation et Evaluation d'une solution Cloud hybride pour les services télécoms

3.1 Introduction

Dans le domaine des télécommunications, les entreprises vendent leurs services à travers des agences commerciales au niveau de chaque wilaya, pour ces agences il est impératif de mettre en place des solutions efficaces pour gérer les tâches,les services offerts et la satisfaction des clients dans le but d'augmenter les gains de l'entreprise.
Le problème c'est qu'avec une architecture cloud traditionnelle (centralisée) les besoins de l'entreprise ne seront pas réalisable a cause des limites de l'architecture notamment la haute latence et la congestion de la bande passante c'est pour cela qu'il faut opter a utiliser une solution pour garantir le bon fonctionnement des services des agences.
Dans ce chapitre nous allons détailler les étapes de la conception et l'implémentation de la solution, les technologies utilisées, puis évaluer cette solution.

3.2 Proposition

Notre proposition se compose de deux tâches principales :

1. **La supervision de la satisfaction des clients :** consiste à surveiller les délais d'attente et de traitement des clients, en utilisant un ensemble de données (dataset) qui enregistre le temps d'arrivée des clients, le temps de début et fin de leur

tour, la durée d'attente du client, la durée de traitement de service, le type de service et le numéro de guichet. En appliquant des algorithmes de machine learning, nous pourrons identifier les anomalies qui surviennent au cours de la journée afin de remplacer l'être humain par un modèle de ML pour cette tâche.

2. **La gestion des ventes :** cette tâche se divise en deux parties :

   - **La classification des clients :** la première partie de cette tâche consiste à analyser les différentes informations collectées sur les clients, telles que le type d'abonnement, la fréquence d'achat, la durée d'abonnement en mois et les revenus mensuels générés par chaque client pour l'entreprise. L'objectif est de classifier ces clients en segments à l'aide de modèles de machine learning.
   - **La prédiction des ventes :** dans la deuxième partie, nous utiliserons un autre dataset contenant des informations détaillées pour chaque transaction. Nous étudierons et traiterons ces données à l'aide d'un autre type de modèles ML, en analysant les types de produits et les quantités vendues sur une longue période. Ces modèles permettront de prédire les ventes totales. Enfin, une comparaison sera effectuée entre les ventes totales réelles et les ventes prédites afin de déterminer l'exactitude des modèles. Pour les trois parties, l'objectif est de sélectionner le modèle le plus performant dans chacune d'elles afin de l'intégrer dans notre architecture.

3.2.1 Architecture proposée :

Pour la conception de ces taches, nous proposons l'adoption d'une architecture cloud hybride avec trois couches : l'edge computing, le fog computing et le cloud computing.

1. **Couche edge computing (au niveau de l'agence) :** ce niveau représente les traitements effectués directement au sein des agences commerciales. Les principales tâches réalisées sont :
   - Suivre les services effectués au niveau de l'agence (paiement, consultation, support, activation).
   - Surveillance en temps réel des temps d'attentes et temps de service.
   - Détection d'anomalies en temps réel.
   - Pré-traitement des données pour les envoyer au niveau fog.
2. **Couche fog computing (niveau wilaya) :** ce niveau intermédiaire assure l'agrégation et l'analyse des données provenant de toutes les agences d'une wilaya. Ses responsabilités incluent :
   - Agrégation et analyse des données de plusieurs agences pour des décisions régionales.
   - Modèles de prédiction pour la gestion des stocks et des ressources (prédire les prochaines ventes pour mieux gérer le stock et l'équilibre entre les matériaux vendus).
   - Synchronisation des données avec le cloud.
3. **Couche cloud computing (niveau national) :** représentant le niveau le plus haut (la direction générale). Ses fonctions sont :
   - Stockage sécurisé et durable de l'historique complet des données.
   - Analyse par machine learning de données collectées.
   - Modélisation prédictive pour anticiper les tendances futures et les pics nationaux.
   - Échange de données avec les niveaux inférieurs : envoi de prévisions/recommandations au fog, et mises à jour des modèles ML à l'edge.

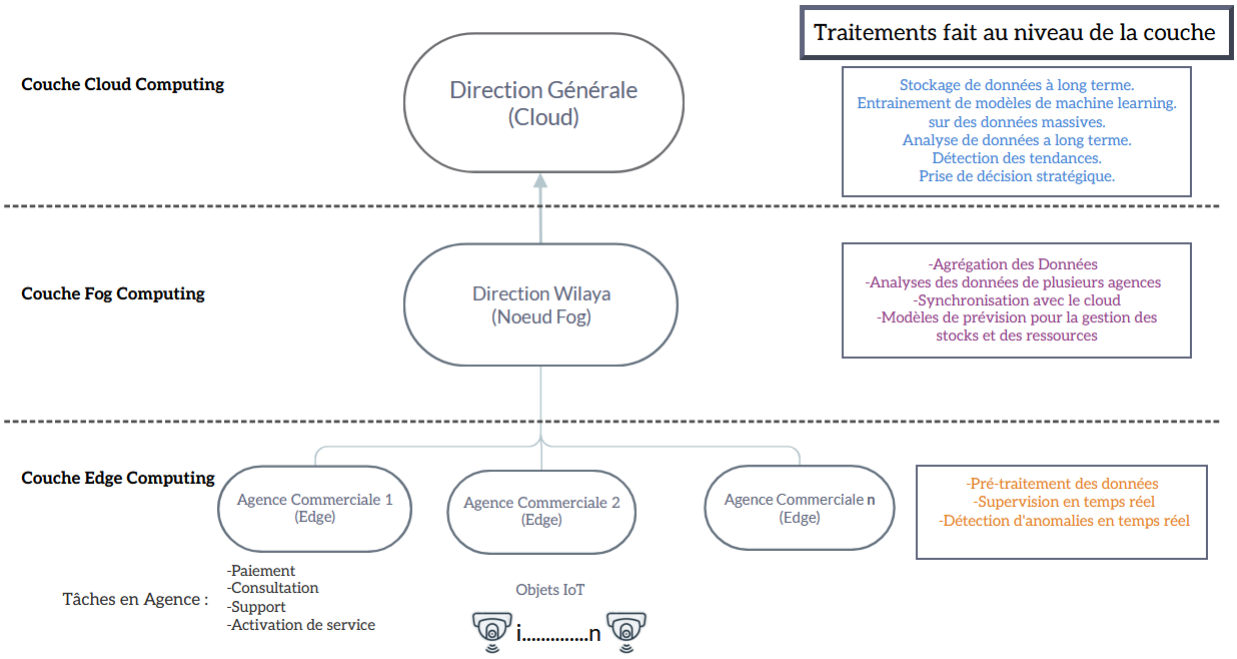

FIGURE 3.1 – Architecture hybride (Edge/Fog/cloud) proposée pour la gestion des services télécoms dans un environnement IoT.

La figure 3.1 illustre les trois niveaux de notre architecture cloud hybride, montrant les traitements effectués à chaque niveau pour intégrer à la fois la supervision de la satisfaction des clients et la gestion des ventes.

3.2.2 Environnement de développement et outils utilisés

1. **Anaconda et jupyter notebook :**
   - **Anaconda :** est une distribution open source des langages de programmation Python et R, utilisée principalement pour la science des données, le machine learning et le calcul scientifique. Elle est développée par Anaconda Inc. et comprend un gestionnaire de paquets et d'environnements appelé conda[1].
   - **Jupyter notebook :** est un outil open source intégré à Anaconda, qui permet de créer et partager des documents interactifs contenant du code live, des visualisations, des équations mathématiques, du texte narratif et plus encore[6].
2. **Librairies Python utilisées**

- **NumPy (Numerical Python) :** Bibliothèque fondamentale pour le calcul scientifique en Python. Elle fournit un support pour les tableaux et matrices multidimensionnelles, ainsi qu'un large éventail de fonctions mathématiques pour travailler avec ces structures de données.
- **Pandas :** Bibliothèque de manipulation et d'analyse de données. Elle offre des structures de données et des outils d'analyse de données performants et faciles à utiliser. Pandas facilite l'importation, le nettoyage, la transformation et la manipulation de données structurées (séries temporelles, données tabulaires, etc.).
- **Scikit-learn :** Bibliothèque d'apprentissage automatique (machine learning) en Python. Elle propose une vaste gamme d'algorithmes d'apprentissage supervisé et non supervisé, ainsi que des outils pour la préparation des données, la sélection des modèles, l'évaluation des performances, etc.
- **Matplotlib :** Bibliothèque de visualisation 2D et 3D en Python. Elle permet de créer des graphiques statiques, histogrammes, diagrammes à barres, graphiques linéaires, graphiques à dispersions, etc. Matplotlib est souvent utilisée en conjonction avec NumPy et Pandas.
- **Seaborn :** Bibliothèque de visualisation de données basée sur Matplotlib. Elle fournit une interface de plus haut niveau pour créer des graphiques statistiques attrayants et informatifs. Seaborn est particulièrement adaptée pour explorer et comprendre les relations entre les variables des données.
- **Random :** Module Python intégré pour générer des nombres aléatoires. Il offre des fonctions pour générer des nombres aléatoires selon différentes distributions de probabilité (uniforme, normale, exponentielle, etc.), ainsi que des fonctions d'échantillonnage[6].

3.3 Modélisation, algorithme, expérimentation et résultats

3.3.1 Modélisation :

Dans une architecture cloud hybride (cloud, fog, edge) il est préférable d'utiliser le protocole MQTT avec Transport Layer Security (TLS) pour les transmissions entre les différents niveaux de l'architecture ce choix repose sur plusieurs avantages clés du protocole :
La Légèreté, l'efficacité, la qualité de service, la scalabilité et la sécurite des communications.

Pour assurer une transmission rapide et fiable des grandes quantités de données échanger entre les niveaux fog et le niveau cloud, l'utilisation des liaisons de fibre optique est recomandée.

3.3.2 Génération de données

Grâce à l'architecture hybride cloud/fog/edge que nous avons proposée, un système de collecte et de traitement des données pro- venant des agences commerciales pourra être déployé. Cependant, comme nous ne disposons pas des moyens nécessaires pour effectuer la collecte de données réelles et les tests sur site, nous avons généré des ensemble de données (dataset) synthétiques représentatives des opérations typiques dans ces agences.

- **Dataset 1** : le premier dataset contient la liste des opérations effectués dans l'agence durant une année 10000 opérations(lignes), le but de ce dataset est d'analyser les transactions en termes de temps d'attente et de service, de type de service, et de détecter d'éventuelles anomalies dans les transactions. Cela sera utile pour réduire les temps d'attente, et améliorer l'efficacité globale des guichets en réduisant le temps de service le plus possible. Voici une description des colonnes :

  – TransactionID : Identifiant unique de la transaction.

- ArrivalDate : Date et heure d'arrivée du client.
- StartDate : Date et heure de début du service.
- FinishDate : Date et heure de fin du service.
- WaitTime : Temps d'attente (en minutes) avant le début du service.
- ServiceTime : Temps de service (en minutes).
- ServiceType : Type de service (consultation, paiement, support, activation).
- GuichetID : Identifiant du guichet où le service a été fourni.
- NumPeople : Nombre de personnes dans la file d'attente.
- Anomaly : Indicateur d'anomalie (0 pour normal, 1 pour anomalie).

	TransactionID	ArrivalDate	StartDate	FinishDate	WaitTime	ServiceTime	ServiceType	GuichetID	NumPeople	Anomaly
0	1	2023-09-04 03:36:00	2023-09-04 03:55:38.394357972	2023-09-04 04:19:19.978962564	19.639906	23.693077	consultation	B	3	0
1	2	2023-01-18 21:03:00	2023-01-18 21:06:03.571709880	2023-01-18 21:57:27.756977784	3.059528	51.403088	payment	B	8	0
2	3	2023-05-13 05:05:00	2023-05-13 05:11:23.063878938	2023-05-13 05:58:02.471327244	6.384398	46.656791	support	A	4	0
3	4	2023-05-28 01:29:00	2023-05-28 01:45:19.654901292	2023-05-28 02:22:38.216161752	16.327582	37.309354	activation	C	7	0
4	5	2023-12-03 19:43:00	2023-12-03 19:52:52.157278236	2023-12-03 20:19:26.311719792	9.869288	26.569241	consultation	B	8	0

FIGURE 3.2 – En-tête du dataset 1.

Les graphes suivants fournissent une vue d'ensemble complète et détaillée sur les transactions qui se trouvent sur le dataset 1 :

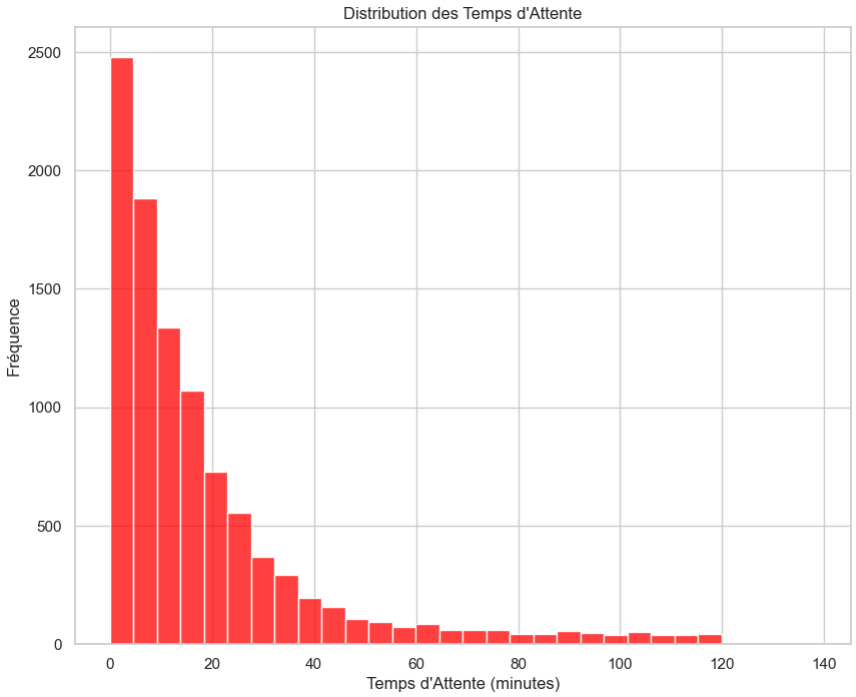

FIGURE 3.3 – La distribution des temps d'attente des transactions en minutes.

- **Axe horizontal :** Représente le temps d'attente.
- **Axe vertical :** Représente la fréquence, c'est-à-dire le nombre de transactions correspondant à chaque intervalle de temps d'attente.

Dans le graphe 3.3 on trouve que la majorité des transactions ont un temps d'attente compris entre 0 et 20 minutes. On observe un pic très marqué autour de 0 à 10 minutes, indiquant que la plupart des clients n'attendent pas longtemps. Au-delà de 20 minutes, la fréquence des transactions diminue progressivement. Il y a très peu de transactions avec un temps d'attente supérieur à 60 minutes.

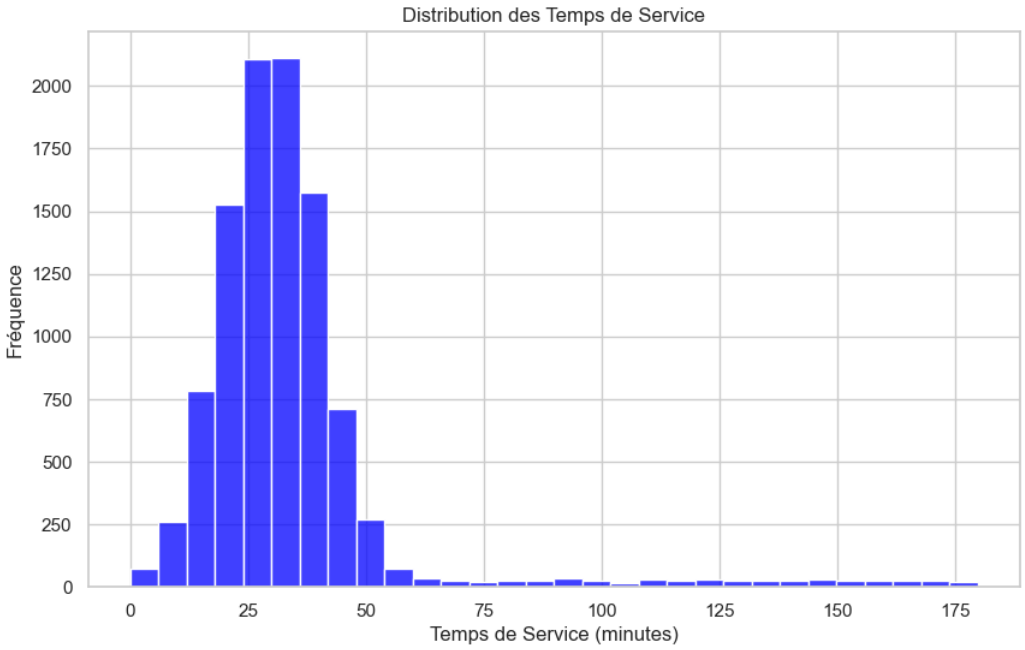

FIGURE 3.4 – La distribution des temps de service des transactions.

- **Axe horizontal :** Représente le temps de service.
- **Axe vertical :** Représente la fréquence, c'est-à-dire le nombre de transactions correspondant à chaque intervalle de temps de service.

Dans le graphe 3.4 on trouve que la majorité des transactions ont un temps de service compris entre 10 et 35 minutes. On observe un pic très marqué autour de 20 à 30 minutes, indiquant que la plupart des services prennent environ ce temps, il existe quelques transactions avec des temps de service exceptionnellement longs, mais elles sont rares.

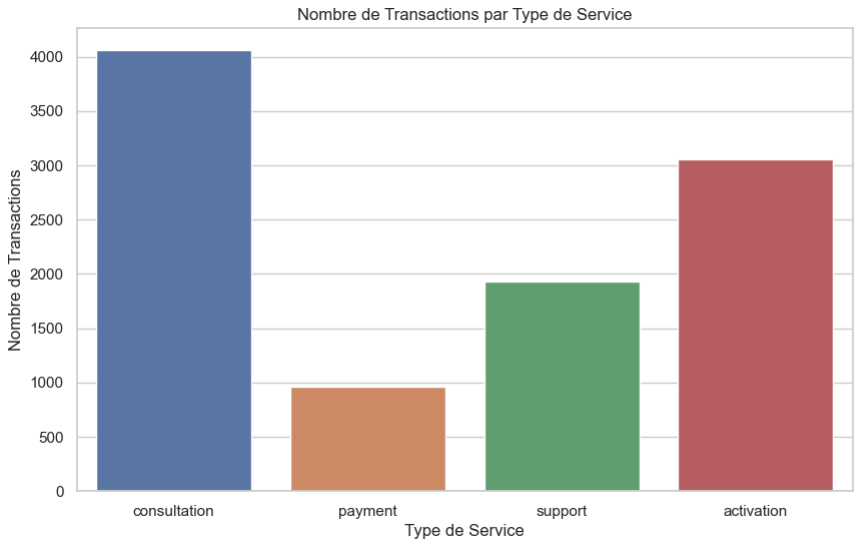

FIGURE 3.5 – Nombre de transactions par type de service.

Le graphe 3.5 fournit une vue d'ensemble claire des préférences et des besoins des clients, on note que :

- La consultation est de loin le service le plus sollicité, ce qui pourrait indiquer que les clients ont souvent besoin d'informations ou de conseils.
- Les services de support et d'activation sont également importants, montrant une demande substantielle pour ces types de services.
- Paiement est le moins fréquent(vu que le paiement en ligne est disponible)

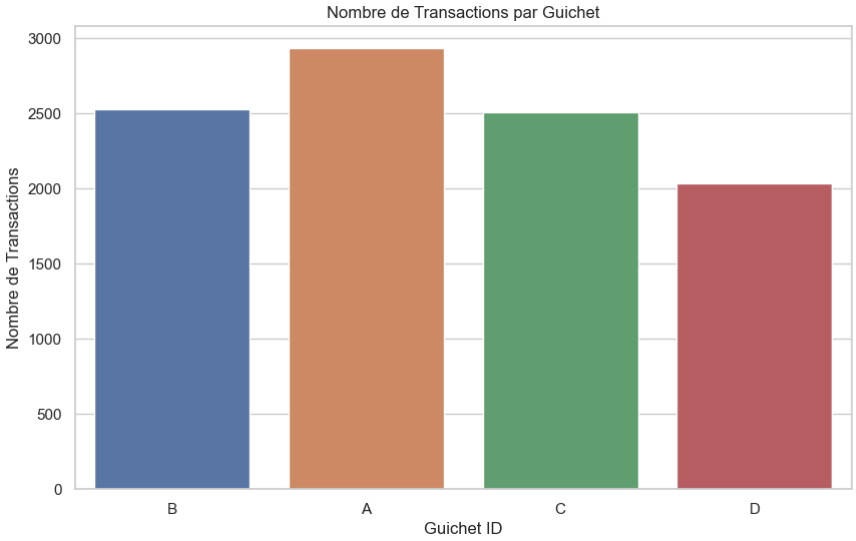

FIGURE 3.6 – Nombre de transactions par guichet.

Le graphe ci-dessus 3.6 montre le nombre de transactions par guichet, révélant que le guichet A est le plus fréquenté, suivi de près par les guichets B et C, avec le guichet D étant le moins utilisé.

- **Dataset 2** : le deuxième dataset contient des informations sur 1000 clients(lignes), réparties sur 8 colonnes.
  - ClientID : chaque client est identifié de manière unique par un identifiant.
  - Age : l'âge des clients.
  - MonthlyRevenue : les revenus que l'entreprise gagne a partir de ce client.
  - PurchaseFrequency : la fréquence des achats montre à quelle fréquence les clients effectuent des transactions.
  - SatisfactionScore : le score de satisfaction donne une idée de la satisfaction globale des clients.
  - SubscriptionType : le type d'abonnement peut être ,"Prepaid","Postpaid" ou "Professional", indiquant les différents types d'abonnement.
  - SubscriptionDuration : la durée de l'abonnement en mois montre combien de temps les clients sont abonnés.
  - Segment : les clients sont classés en différents segments de marché.

	ClientID	Age	MonthlyRevenue	PurchaseFrequency	SatisfactionScore	SubscriptionType	SubscriptionDuration	Segment
0	1	56	6982	14	2	Postpaid	24	2
1	2	69	2006	4	4	Postpaid	57	1
2	3	46	3186	6	2	Professional	58	1
3	4	32	8252	19	5	Postpaid	5	0
4	5	60	5850	6	1	Postpaid	6	1

FIGURE 3.7 – En-tête du dataset 2.

Voici des graphes et anneaux qui illustrent les informations importantes du dataset 2 :

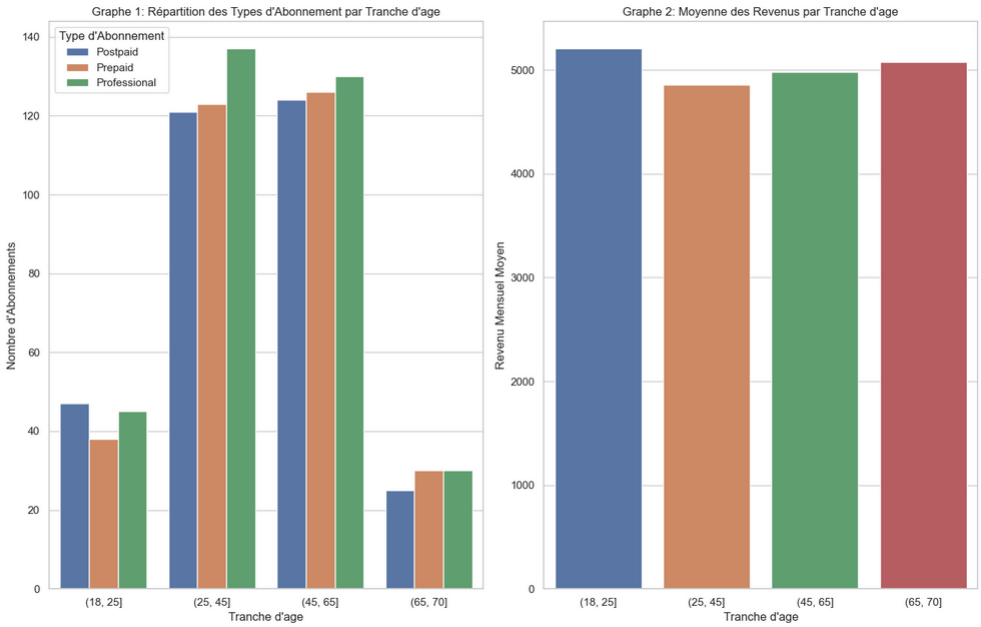

FIGURE 3.8 – Répartition des types d'abonnement et moyenne des revenus par tranche d'age.

- Dans le premier graphe de la figure 3.8, on trouve que :
  - La tranche d'âge [25, 45[a le plus grand nombre d'abonnements pour tous les types, suivie par la tranche [45,65[.
  - Le type d'abonnement "Professional" est légèrement plus populaire dans les tranches d'âge [25, 45[et [45, 65[.
- Pour le deuxième graphe de la figure 3.8 on note :
  - Les clients de la tranche d'âge [18, 25[ont le revenu mensuel moyen le plus élevé pour l'entreprise, supérieur à 5000.
  - Les revenus mensuels moyens sont relativement similaires pour les tranches d'âge [25, 45[, [45, 65[, et [65, 70[, avec des valeurs autour de 5000.

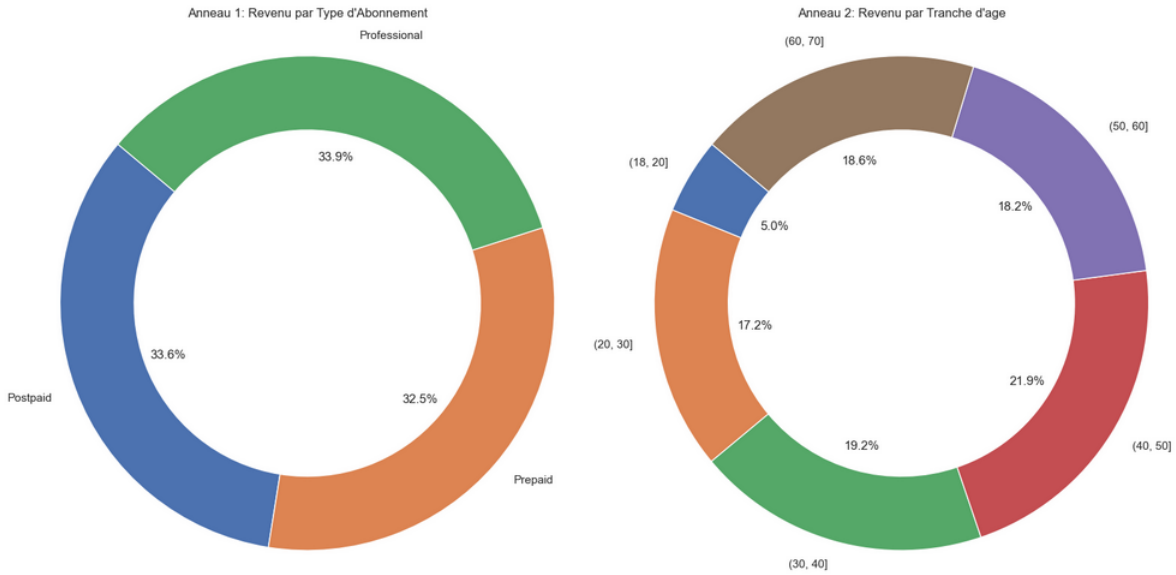

FIGURE 3.9 – Revenus par Type d'abonnement et tranche d'age.

- Le premier anneau de la figure 3.9 montre les revenus de l'entreprise par type d'abonnement :
  - Les revenus sont relativement bien répartis entre les trois types d'abonnement.
  - Le type d'abonnement "Professional" génère légèrement plus de revenus 33.9%.
  - Les types "Postpaid" et "Prepaid" ont des parts de revenus très similaires, autour de 33.6% et 32.5%.
- Le deuxième anneau de la figure 3.9 montre les revenus par tranche d'age, contrairement au graphes en haut les tranches d'age commence par [18,20[et puis pour chaque tranche on prends 10 ans :
  - la tranche d'âge [40, 50[génère la plus grande part du revenu total, avec 21.9%.
  - Les tranches d'âge [30, 40[, [50, 60[, et [60, 70] ont des parts de revenu similaires, autour de 18% 19%.
  - La tranche d'âge [18, 20[génère la plus petite part du revenu total, avec seulement 5%. La répartition des revenus est relativement équilibrée entre les tranches d'âge, à

l'exception de la tranche [18, 20[qui est significativement plus faible.

- **Dataset 3 :** contient les informations des 30000 transactions(lignes) effectués durant deux ans. Ce dataset est réparti en 8 colonnes.

  La figure 3.10 montre l'en-tête du dataset 3 où chaque ligne représente une transaction, les différents attributs présents dans l'en-tête sont :

  – TransactionID : Identifiant unique de la transaction.
  – TransactionDate : Date et heure de la transaction.
  – ClientID : Identifiant unique du client.
  – Quantity : Quantité de produits achetés.
  – UnitPrice : Prix unitaire du produit.
  – Region : Région où la transaction a eu lieu.
  – ProductType : Type de produit acheté.
  – TotalSales : Total des ventes.

Out[153]:

	TransactionID	TransactionDate	ClientID	Quantity	UnitPrice	Region	ProductType	TotalSales
0	1	2022-12-29 06:00:00	103	47	901.428895	East	ZTE Modem	42367.158073
1	2	2023-12-19 07:00:00	436	71	638.662313	South	ZTE Modem	45345.024231
2	3	2022-07-17 15:00:00	861	72	892.627445	North	Huawei Modem	64269.176053
3	4	2023-02-07 00:00:00	271	42	936.108110	North	Repeter	39316.540626
4	5	2023-03-09 08:00:00	107	51	679.347209	North	ZTE Modem	34646.707656

FIGURE 3.10 – En-tête du dataset 3.

3.3.3 Algorithmes du machine learning :

Trois types d'algorithmes de ML ont été employés dans cette étude :

3.3.3.1 Algorithmes de détection d'anoamlies :

Le dataset 1 contient des informations complexes où des anomalies peuvent se manifester sous forme de valeurs aberrantes

ou de comportements inhabituels. Pour identifier l'approche la plus performante sur ce jeu de données, nous appliquons une série d'algorithmes de détection d'anomalies reconnus. Spécifiquement, nous utilisons Isolation Forest, DBSCAN, One-Class SVM, et LOF pour analyser le dataset 1. L'objectif est de comparer leurs performances et de sélectionner celui qui produit les meilleurs résultats en termes de détection d'anomalies sur ce dataset particulier.

3.3.3.2 Algorithmes de classification :

Le dataset 2 comprend des instances de données avec des étiquettes de classe clairement définies. Chaque instance appartient à une catégorie spécifique, ce qui rend les algorithmes de classification particulièrement adaptés pour apprendre à partir de ces étiquettes et effectuer des prédictions précises sur de nouvelles données. Pour identifier l'approche la plus performante sur ce jeu de données, nous appliquons une série d'algorithmes de classification reconnus. Spécifiquement, nous utilisons K-Nearest Neighbors, Decision Tree, Random Forest et SVM pour analyser le dataset 2. L'objectif est de comparer leurs performances et de sélectionner celui qui produit les meilleurs résultats en termes de classification sur ce dataset particulier.

3.3.3.3 Algorithmes de prédiction :

le dataset 3 contient des transactions effectuées pendant deux ans. Chaque transaction inclut des informations telles que la date, le montant, le type de produit, et d'autres variables pertinentes. Pour identifier l'approche la plus performante sur ce jeu de données, nous appliquons une série d'algorithmes de prédiction reconnus. Spécifiquement, nous utilisons régression linéaire, réseau de neurones et random forest regressor pour analyser le dataset 3. L'objectif est de comparer leurs performances et de sélectionner celui qui produit les meilleurs résultats en termes de prédiction sur ce dataset particulier.

3.3.4 Expérimentation et résultats

Après l'analyse exploratoire des données, l'étude procède à l'application des modèles du ML.

3.3.4.1 Application des modèles de ML :

1. **Dataset 1 :** le dataset est initialement pré-traité en faisant le Chargement des données et séparation des caractéristiques (features) et les labels (étiquettes). Puis, quatre modèles de détection d'anomalies sont entraînés sur cet ensemble de données. Enfin, l'évaluation des performances est réalisée. Voici les tests effectués pour chaque modèle :

   - **Modèle IsolationForest :** Possède deux paramètres 'random_state' et 'contamination' qui jouent des rôles importants dans la façon dont le modèle est entraîné et comment il détecte les anomalies :
     - **Random_state :** est utilisé pour contrôler la reproductibilité des résultats lors de l'entraînement du modèle.En fixant une valeur spécifique pour random_state (dans notre cas c'est 42), les résultats de l'entraînement du modèle seront les mêmes à chaque exécution du code avec la même valeur de random_state. Cela est utile pour garantir la cohérence des résultats et faciliter la comparaison entre différentes exécutions du modèle.
     - **Contamination :** définit la proportion anticipée d'anomalies dans le jeu de données. En spécifiant cette valeur, on informe le modèle sur la quantité d'anomalies qu'il devrait chercher à détecter. Une valeur plus élevée de contamination conduit le modèle à rechercher un plus grand nombre d'anomalies, tandis qu'une valeur plus faible le restreint à identifier uniquement les anomalies les plus flagrantes pour notre cas on va

tester et comparer avec trois valeurs de contamination (0.01-0.05-0.1).

- **Modèle OneClassSVM :** possède un paramètre 'nu' qui représente la proportion d'échantillons d'anomalies attendue dans l'ensemble de données.En ajustant le paramètre nu, vous pouvez contrôler la flexibilité du modèle dans la délimitation entre les données normales et les anomalies. Une valeur plus faible de nu permet au modèle d'être plus strict dans la détection des anomalies, tandis qu'une valeur plus élevée de nu permet au modèle d'être plus tolérant envers les anomalies on va tester avec trois valeurs de nu comme avant.
- **Modèle DBSCAN :** Les deux paramètres clés de ce modèle sont 'eps' et 'min_samples',eps contrôle la taille du voisinage autour de chaque point,les points situés à une distance inférieure ou égale à eps sont considérés comme voisins et sont regroupés ensemble, tandis que min_samples détermine le nombre minimum de points nécessaires pour former un cluster,pour qu'un point soit considéré comme un point central d'un cluster, il doit avoir au moins min_samples voisins dans un rayon de eps les points qui ne satisfont pas cette condition sont considérés comme des anomalies.
  Ces deux paramètres sont essentiels pour ajuster la sensibilité de DBSCAN à la densité des données et pour détecter efficacement les clusters et les anomalies.
  Pour nos tests, nous commencerons avec un eps de 0.1 et augmenterons de 0.1 à chaque test jusqu'à atteindre la plus haute précision possible, en maintenant min_samples à 5 pour tous les tests.
- **Modèle LocalOutlierFactor :** LOF est instancié avec deux paramètres principaux :
  'n_neighbors' qui est le nombre de voisins à considérer pour évaluer la densité locale de chaque point et 'con-

tamination' qui est la proportion attendue d'anomalies dans les données.
Pour nos expérimentations, la valeur de contamination est maintenue à 0.05. Nous commençons avec n_neighbors=10 et augmentons progressivement cette valeur jusqu'à obtenir une précision optimale.

2. **Dataset 2 :** quatres modèles de classification sont entrainés sur le dataset.Ensuite,les performances de chaque modèle sont évaluées pour déterminer le meilleur modèle pour ce dataset, voici les tests :

   - **K-Nearest Neighbors :** possède un seul paramètre clé "n_neighbors" qui est le nombre de voisins à considérer pour la classification. Un nombre plus élevé de voisins peut rendre le modèle plus robuste mais moins sensible aux variations locales, pour ce test la valeur de "n_neighbors" est 3.
   - **Decision Tree :** possède un seul paramètre clé "max_depth" qui signifie la profondeur maximale de l'arbre. Une profondeur plus grande permet à l'arbre de capturer plus de détails, mais augmente également le risque de surapprentissage. Pour ce test "max_depth=9".
   - **Random Forest :** possède un seul paramètre clé "max_depth" qui est la profondeur maximale des arbres dans la forêt, Une grande profondeur permet de capturer plus de détails mais augmente le risque de surapprentissage. Le test est effectué avec "max_depth=10".
   - **Support Vector Machine (SVM) :** possède un seul paramètre clé "C" qui est le paramètre de régularisation. Un "C" plus grand signifie moins de régularisation, ce qui permet au modèle de mieux s'adapter aux données d'entraînement mais augmente le risque de surapprentissage. La valeur de "C" dans ce test est : 20.

Les performances des modèles entrainés sur les dataset 1 et 2 seront évaluées en mesurant l'exactitude(accuracy_score) et le rapport de classification(classification_report).

3. **Dataset 3 :** pour ce dataset 3 modèles de prédiction sont entrainés sur les caractéristiques du dataset pour prédire les ventes. Ensuite, leurs performances sont évalués en mesurant le "MAE"(Erreur Absolue Moyenne) et le coefficient de détermination "R^2" pour déterminer le meilleur modèle pour ce dataset.

    - **MAE :** est une mesure de l'erreur moyenne entre les valeurs observées et les valeurs prédites. Il est calculé en prenant la moyenne des valeurs absolues des écarts entre les prédictions et les valeurs réelles. Plus le MAE est faible, meilleure est la performance du modèle.
    - **R^2** : également appelé R-squared, est une mesure qui indique la proportion de la variance des variables dépendantes qui est expliquée par le modèle. Il varie de 0 à 1 et plus sa valeur est proche de 1, meilleure est l'adéquation du modèle aux données. Un R^2 de 1 signifie que les valeurs prédites par le modèle correspondent exactement aux valeurs réelles.

3.3.4.2 Résultats :

nous présenterons ici les résultats des tests des modèles appliqués aux 3 dataset qui permettront de comparer l'efficacité des différents modèles et de déterminer celui qui offre les meilleures performances pour chaque dataset.

1. **Dataset 1 :** Le nombre d'anomalies dans ce dataset est 500, voici les résultats des performances de ces modèles :

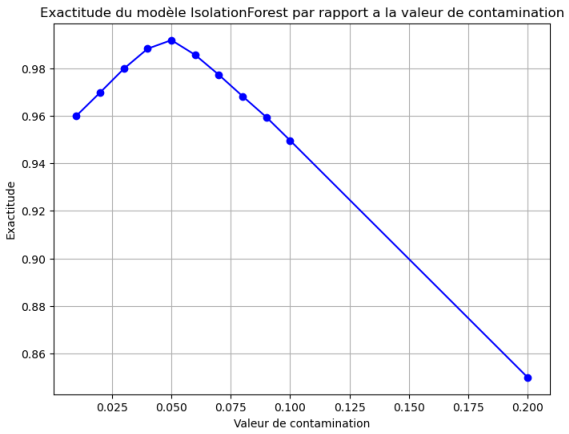

FIGURE 3.11 – Exactitude de IsolationForest.

Le graphique 3.11 illustre l'exactitude du modèle Isolation Forest en fonction de la valeur de contamination. Le graphique montre que la meilleure exactitude (99.1%) est obtenue avec une valeur de contamination de 0.05. Il est clair qu'au-delà d'une valeur de contamination de 0.05, l'exactitude diminue de manière significative.

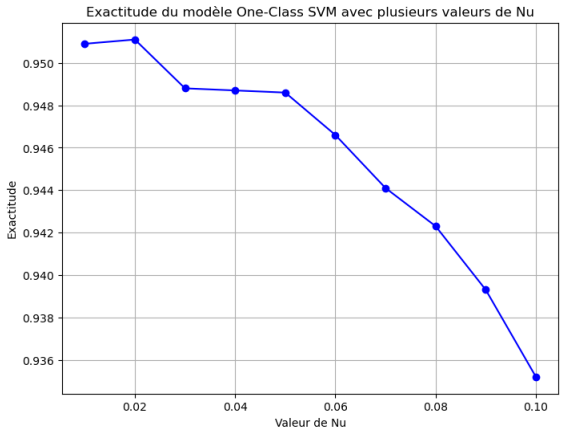

FIGURE 3.12 – Exactitude de OneClassSVM.

Le graphe 3.12 montre qu'avec le modèle One-Class SVM la plus haute exactitude possible est 95% avec une valeur de Nu=0.02 on remarque qu'a chaque incrémentation de la valeur Nu l'exactitude se diminue jusqu'elle arrive a (exactitude= 93.52%, Nu= 0.1).

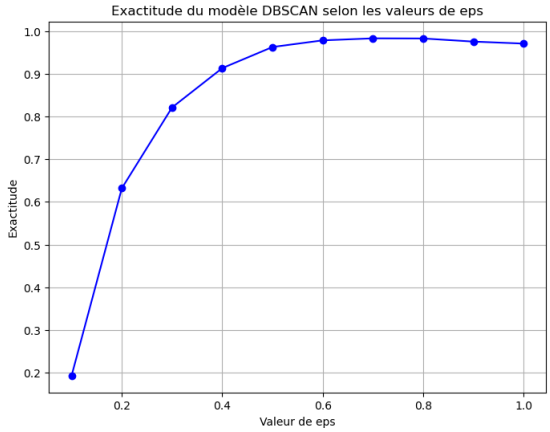

FIGURE 3.13 – Exactitude de DBSCAN .

Le graphe 3.13 montre qu'avec le modèle DBSCAN,l'exactitude fortement lorsque la valeur de eps passe de 0.1 à environ 0.5.
De eps = 0.5 à environ eps = 0.7, il y a une augmentation plus lente jusqu'à ce qu'elle atteigne la plus haute exactitude (98%) avec eps=0.7 .

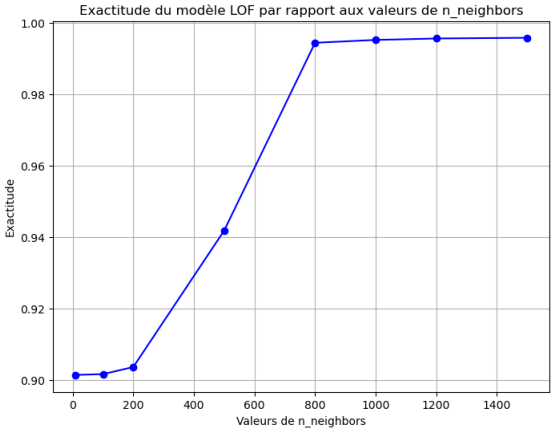

FIGURE 3.14 – Exactitude de LOF.

Le graphique 3.14 représente l'exactitude d'un modèle Local Outlier Factor (LOF) en relation avec différentes valeurs du paramètre n_neighbors a partir du graphique on observe :

- Pour des valeurs faibles de n_neighbors <100 l'exactitude est également faible.
- À mesure que la valeur de n_neighbors augmente, on ob-

serve une augmentation importante de l'exactitude entre 200 et 800.

- L'exactitude continue d'augmenter à un rythme plus lent entre 800 et 1400.
- Enfin, le graphique se stabilise avec une exactitude maximale 99.5% a partir de 1400.

Modèle	Exactitude	Précision	Rappel	Score F1	Support
IsolationForest	99.1%	99%	99%	99%	500
OneClassSVM	95%	94%	95%	94%	500
DBSCAN	98%	98%	98%	98	500
LOF	99.58%	99%	99%	99%	500

TABLEAU 3.1 – Performance des modèles de détection d'anomalies.

Précision : C'est le ratio des vrais positifs sur le total des prédictions positives.
Recall : Ou rappel c'est le ratio des vrais positifs sur le total des éléments réellement positifs. Elle mesure la capacité du modèle à identifier correctement les échantillons positifs.
Score F1 : C'est la moyenne de la précision et du rappel

D'aprés les évaluations des performances de ces modèles sur le **dataset 1**, on trouve que le ***LOF*** est le plus précis avec une exactitude de 99.5%, ensuite ***IsolationForest*** avec 99.1%,puis ***DBSCAN*** avec 98%, et au dernier ***OneClassSVM*** avec une exactitude de 95%.

2. **Dataset 2 :** Il y a un total de 200 données à classer, réparties en trois classes (0, 1, 2) avec les répartitions suivantes : la classe 0 comprend 71 exemples, la classe 1 comprend 71 exemples, et la classe 2 comprend 58 exemples,voici les résultats trouvés :

```
Exactitude du modèle KNN : 0.99
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        71
           1       0.99      1.00      0.99        71
           2       1.00      0.98      0.99        58

    accuracy                           0.99       200
   macro avg       1.00      0.99      0.99       200
weighted avg       1.00      0.99      0.99       200
```

FIGURE 3.15 – Exactitude du modèle K-Nearest Neighbors.

La figure 3.15 montre que l'exactitude du modèle KNN est de 0.99, ce qui signifie que le modèle a correctement prédit 99% des cas lors de l'évaluation. Il s'agit d'une performance très élevée.

```
Modèle Decision Tree:

max_depth: 9
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        71
           1       1.00      1.00      1.00        71
           2       1.00      1.00      1.00        58

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

FIGURE 3.16 – Exactitude du modèle arbre de décision.

La figure 3.16 montre que le modèle arbre de décision a obtenu une exactitude parfaite de 100% avec une profondeur maximale de 9, ce qui signifie qu'il a correctement classifié tous les données du test.

```
Modèle RandomForest:

max_depth: 10
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   suppor

           0       1.00      1.00      1.00        7
           1       1.00      1.00      1.00        7
           2       1.00      1.00      1.00        5

    accuracy                           1.00       26
   macro avg       1.00      1.00      1.00       26
weighted avg       1.00      1.00      1.00       26
```

FIGURE 3.17 – Exactitude du modèle RF.

```
Modèle SVM: C= {20}
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        71
           1       1.00      1.00      1.00        71
           2       1.00      1.00      1.00        58

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```

FIGURE 3.18 – Exactitude du modèle SVM.

La figure 3.17 montre que le modèle Random Forest a ob-

tenu une exactitude parfaite de 100% avec une profondeur maximale de 10.

La figure 3.18 montre que le modèle SVM a obtenu une exactitude parfaite de 100% avec un "C" de 20.
Étant donné les résultats obtenus, pour ce dataset il est possible de choisir n'importe quelle modèle parmi les trois qui ont eu une exactitude de 100% .

3. **Dataset 3 :** les résultats obtenus sont :

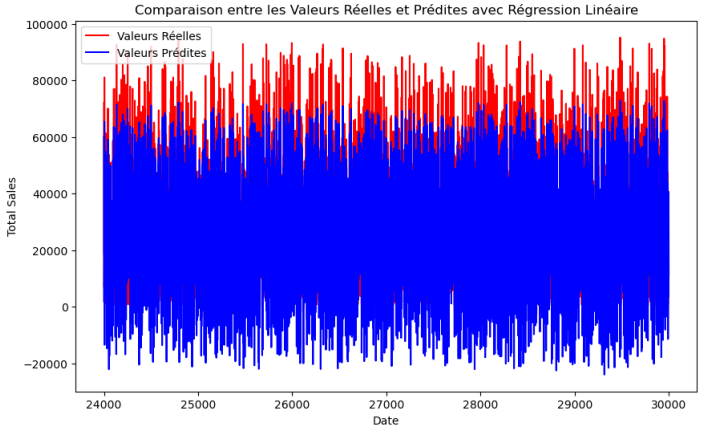

FIGURE 3.19 – Graphe du modèle regression linéaire.

La figure 3.19 montre que les valeurs réelles et prédites suivent une tendance similaire, mais il y a des écarts plus prononcés, en particulier pour les valeurs extrêmes. Les valeurs prédites (bleu) ont une dispersion plus large par rapport aux valeurs réelles (rouge), cela indique que ce modèle capture la tendance générale mais il manque de précision surtout pour les valeurs extrêmes. Il y a des erreurs de prédiction plus importantes comparées aux deux autres modèles.

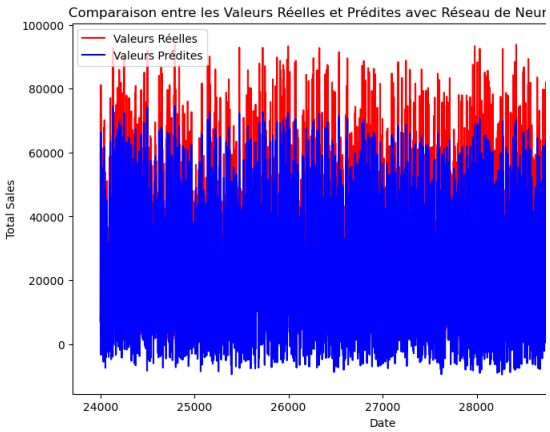

FIGURE 3.20 – Graphe du modèle RN.

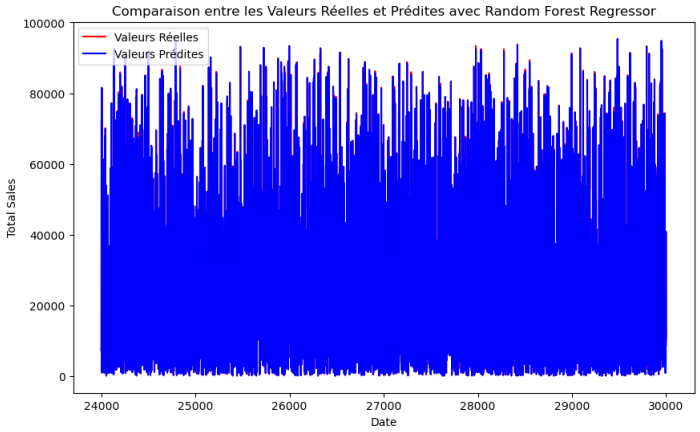

FIGURE 3.21 – Graphe du modèle RFR.

La figure 3.20 montre que les valeurs réelles et prédites semblent suivre une tendance similaire, mais il y a des écarts notables entre les deux. Les valeurs réelles (rouge) ont des pics plus élevés que les valeurs prédites (bleu) à plusieurs endroits, cela signifie que ce modèle capture globalement la tendance des données, mais manque de précision dans certains cas, surtout pour les valeurs les plus élevées.

La figure 3.21 montre que les valeurs prédites par le Random Forest Regressor sont très proches des valeurs réelles, avec très peu de différences visibles. Cela signifie que ce modèle est très précis dans ses prédictions, capturant presque parfaitement les variations des ventes totales.

```
Modèle Random Forest Regressor
Mean Absolute Error: 85.563
R² Score: 0.9999605955148935

Modèle régression linéaire :
Mean Absolute Error: 6204.363
R² Score: 0.857

Modèle réseau de neurones:
Mean Absolute Error: 5017.342
R² Score: 0.907
```

FIGURE 3.22 – Résultats de l'évaluation des trois modèles.

La figure 3.22 présente les résultats de performance de trois modèles, les métriques utilisées pour évaluer ces modèles sont le MAE et le R^2 Score. On remarque que :

- Le Random Forest Regressor est le modèle le plus précis avec un MAE très bas (85.56) et un "R^2" Score presque parfait (0.99).
- Le Réseau de Neurones vient en deuxième position avec un MAE de (5017.342) et un "R^2" Score de (0.907), indiquant une bonne performance mais moins précise que le premier.
- La Régression Linéaire est la moins performante des trois, avec un MAE de (6204.363) et un "R^2" Score de (0.857).

Architecture	Infrastructure utilisée	Protocole de sécurité	Modèles utilisés	Outil utilisé
Architecture cloud hybride pour les services télécoms	hybride (Edge/ Fog/ Cloud)	MQTT avec TLS	LOF pour la détection d'anomalies, SVM pour la classification et Random Forest Regressor pour la prédiction	Machine Learning

Tableau 3.2 – Tableau récapitulatif de l'architecture proposée.

3.4 Conclusion

Ce chapitre a présenté une solution cloud hybride pour améliorer la gestion des services télécoms. L'architecture proposée, combinant edge, fog et cloud computing, offre une approche flexible et efficace pour traiter les défis de latence, de bande passante et d'analyse de données en temps réel présents dans les architectures cloud traditionnels. L'utilisation d'algorithmes de machine learning pour la détection d'anomalies, la classification des clients et la prédiction des ventes démontre le potentiel de cette solution pour optimiser les opérations et améliorer la satisfaction des clients.
Les expérimentations menées sur des données synthétiques ont permis d'identifier les avantages potentiels de cette architecture. Cependant, il est important de noter que des tests sur des données

réelles seront nécessaires pour confirmer pleinement l'efficacité de la solution dans un environnement opérationnel.

Conclusion générale et perspectives

Ce mémoire a exploré la conception et l'implémentation d'une architecture cloud hybride pour les systèmes intelligents ambiants, avec une application spécifique aux services de télécommunications dans le contexte de l'IoT. Notre travail a débuté par une analyse approfondie des concepts fondamentaux de l'IoT. Cette base théorique nous a permis de comprendre les enjeux et les défis actuels de ce domaine en pleine expansion. L'état de l'art détaillé sur les architectures Cloud, Fog et Edge Computing a révélé les forces et les limites des approches existantes. La contribution principale de ce mémoire réside dans la proposition d'une architecture combinant les avantages des paradigmes Edge, Fog et Cloud Computing. Cette approche hybride a été spécifiquement conçue pour optimiser la gestion des services télécoms pour dans un environnement IoT pour augmenter les gains de l'entreprise, L'implémentation de cette architecture s'est appuyée sur diverses techniques de machine learning, notamment pour la détection d'anomalies, la classification et la prédiction. Les résultats expérimentaux ont démontré des améliorations significatives en termes de performance et de précision pour diverses tâches d'analyse de données IoT. Pour les travaux futurs, plusieurs axes d'amélioration peuvent être envisagés : L'utilisation des données réelles provenant des agences de plusieurs régions pour valider et généraliser les résultats obtenus. Implémentation et test de la solution dans un environnement réel d'entreprise de télécommunications.

Bibliographie

[1] Anaconda (distribution python). https://fr.wikipedia.org/wiki/Anaconda_(distribution_Python)(consulté le 13/04/2024).

[2] Arima model - complete guide to time series forecasting in python. url = https ://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/(consulté le 26/06/2024).

[3] Differences between cloud, fog and edge computing in iot. https://www.digiteum.com/cloud-fog-edge-computing-iot/(consulté le 11/04/2024).

[4] File management algorithm reference guide. url=https ://docs.rubiscape.com/rarg(consulté le 26/06/2024).

[5] Predictive modeling analytics and machine learning. https://www.sas.com/en_gb/insights/articles/analytics/a-guide-to-predictive-analytics-and-machine-learning.html(consulté le 13/04/2024).

[6] Python data science handbook. https://github.com/jakevdp/PythonDataScienceHandbook(consulté le 13/04/2024).

[7] Qui est ce que la détection des anomalies. https://www.elastic.co/fr/what-is/anomaly-detection(consulté le 11/04/2024).

[8] Qu'est-ce que le clustering ? les 3 méthodes à connaitre. https://larevueia.fr/clustering-les-3-methodes-a-connaitre/(consulté le 12/04/2024).

[9] Unsupervised machine learning. https://www.javatpoint.com/unsupervised-machine-learning(consulté le 26/06/2024). Free learning platform for better future.

[10] A quick and dirty guide to random forest regression. url=https ://towardsdatascience.com/a-quick-and-dirty-guide-to-random-forest-regression(consulté le 26/06/2024), 2020.

[11] Machine learning in iot security : Current solutions and future challenges. https://moncoachdata.com/blog/modeles-de-machine-learning-expliques/(consulté le 11/06/2024), 28 août 2022.

[12] A.A.Abdellatif, A. Mohamed, C.F. Chiasserini, M.Tlili, and A.Erbad. Edge computing for smart health : Context-aware approaches, opportunities, and challenges. *IEEE Network*.

[13] A.R.Nasser, A.M. Hasan, A.J.Humaidi, A.Alkhayyat, L.Alzubaidi, M.A.Fadhel, and Y.Duan. Iot and cloud computing in health-care : A new wearable device and cloud-based deep learning algorithm for monitoring of diabetes. *Electronics*, 2021.

[14] A.T.Atieh. The next generation cloud technologies : a review on distributed cloud, fog and edge computing and their opportunities and challenges. *ResearchBerg Review of Science and Technology*, 2021.

[15] M.De Donno, K.Tange, and N.Dragoni. Foundations and evolution of modern computing paradigms : Cloud, iot, edge, and fog. 2019.

[16] D.Serpanos and M.Wolf. *Internet-of-Things (IoT) Systems*. Springer International Publishing, 2018.

[17] F.Firouzi, K.Chakrabarty, and S.Nassif. *Intelligent internet of things : From device to fog and cloud*. Springer Nature, 2020.

[18] F.Hussain, R.Hussain, S.A.Hassan, and E.Hossain. Machine learning in iot security : Current solutions and future challenges. *IEEE Communications Surveys & Tutorials*, 2020.

[19] H.Chaouchi. *The Internet of Things : Connecting Objects to the Web*. ISTE, 2010.

[20] H.MOUDOUD. *Intégration de la Blockchain à l'Internet des Objets*. Thèse de doctorat, Université de Technologie de Troyes, 2022.

[21] H.Wang, C.Ma, and L.Zhou. A brief review of machine learning and its application.

[22] I.Kertiou. *Un middleware pour la gestion de la sensibilité au contexte dans les systèmes de l'Internet des Objets*. PhD thesis, 2021.

[23] Javatpoint. Cloud computing architecture,. `https://www.javatpoint.com/cloud-computing-architecture`/(consulté le 11/04/2024).

[24] J.M.Keller, M.R.Gray, and J.A.Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*.

[25] K.Dolui and S.K.Datta. Comparison of edge computing implementations : Fog computing, cloudlet and mobile edge computing. In *2017 Global Internet of Things Summit (GIoTS)*. IEEE, 2017.

[26] L.Belcastro, F.Marozzo, A.Orsino, D. Talia, and P.Trunfio. Edge-cloud continuum solutions for urban mobility prediction and planning. 2023.

[27] C.Ge L.Fang, G.Zu, and all. L.fang, c.ge, g.zu, x.wang, w.ding, c.xiao and l.zhao. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, 2019.

[28] Veritas Technologies LLC. Tout ce qu'il faut savoir sur l'edge computing. `https://www.veritas.com/fr/ch/information-center/edge-computing`(consulté le 18/04/2024).

[29] M.Jabeen and K.Ishaq. Internet of things in telecommunications : From the perspective of an emerging market. *Journal of Information Technology Teaching Cases*, 2024.

[30] M.Çelik, F.Dadaşer-Çelik, and A.Ş.Dokuz. Anomaly detection in temperature data using DBSCAN algorithm. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 2011.

[31] R.Mahmud, F.L.Koch, and R.Buyya. Cloud-fog interoperability in iot-enabled healthcare solutions. 2018.

[32] S.Hariri, M.C.Kind, and R.J.Brunner. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[33] Faker Skandrani. Architecture iot : L'essentiel à savoir. `https://iotindustriel.com/iot-iiot/architecture-iot-lessentiel-a-savoir/`(consulté le 11/04/2024), 2024.

[34] S.Mughal, K.S. Awaisi, A. Abbas, I. Ur Rehman, M.U.S Khan, and M.alI. A reliable and efficient fog-based architec-

ture for autonomous vehicular networks. *Fog Computing : Theory and Practice*, 2020.

[35] T.Djemai. *Placement optimisé de services dans les architectures fog computing et internet of things sous contraintes d'énergie, de QoS et de mobilité, year =2021, school =Université Toulouse 3 Paul Sabatier, type =Thèse de doctorat, address=Toulouse, France,.* PhD thesis.

[36] X.Wang and Y.Li. Fog-assisted content-centric healthcare iot. *IEEE Internet of Things Magazine*, 2020.

[37] Y.A.Daraghmi, E.Y.Daraghmi, R.Daraghma, Fouchal, and M.Ayaida. Edge–fog–cloud computing hierarchy for improving performance and security of nb-iot-based health monitoring systems. *Sensors*, 2022.

[38] Y.D.Chen, M.Z.Azhari, and J.S.Leu. Design and implementation of a power consumption management system for smart home over fog-cloud computing. *Department of Electronic and Computer Engineering, Taipei, Taiwan*, 2018.

[39] Y.Liu, S.Bi, Z.Shi, and L.Hanzo. When machine learning meets big data : A wireless communication perspective. *IEEE Vehicular Technology Magazine*, 2019.

Résumé

Ce mémoire propose la conception d'une architecture cloud hy-bride pour les services télécoms commerciaux, combinant les paradigmes edge, fog et cloud computing. L'objectif principalest d'optimiser la gestion des données et l'exécution de tâches d'analyse avancées dans un environnement IoT en exploitant les avantages de chaque couche. Cette approche hybride est ap-pliquée à trois cas d'usage clés dans le domaine des télécoms. Des modèles de machine learning, sont utilisés pour réaliser ces tâches. Les résultats des tests démontrent l'efficacité de l'archi-tecture proposée en termes de performance et de flexibilité, avec des exactitudes presque parfaite. Cette solution offre aux opé-rateurs télécoms une plateforme pour améliorer leurs services commerciaux, optimiser leurs opérations et prendre des déci-sions éclairées basées sur les données.

Mots clés : Cloud computing, IdO, Apprentissage automa-tique, IA.

Abstract

This thesis proposes the design of a hybrid cloud architecture for commercial telecom services, combining Edge, Fog, and Cloud computing paradigms. The main objective is to optimize data management and execute advanced analytics tasks in an IoT environment by leveraging the advantages of each layer. This hybrid approach is applied to three key use cases in the tele-com domain. Machine learning models are used to perform these tasks. Test results demonstrate the effectiveness of the proposed architecture in terms of performance and flexibility. with almost perfect accuracy scores. This solution offers telecom operators a platform to improve their commercial services, optimize their operations, and make data-driven decisions

Keywords : Cloud computing, IoT, machine learning, AI

ملخص

تقترح هذه الأطروحة تصميم بنية سحابية هجينة لخدمات الاتصالات التجارية، تجمع بين نماذج الحوسبة الطرفية والضبابية والسحابية. الهدف الرئيسي هو تحسين إدارة البيانات وتنفيذ مهام التحليل المتقدمة في بيئة إنترنت الأشياء من خلال الاستفادة من مزايا كل طبقة. يتم تطبيق هذا النهج الهجين على ثلاث حالات استخدام رئيسية في مجال الاتصالات. تُستخدم نماذج التعلم الآلي لإنجاز هذه المهام. تُظهر نتائج الاختبارات فعالية البنية المقترحة من حيث الأداء والمرونة، مع دقة تكاد تكون مثالية. يوفر هذا الحل لمشغلي الاتصالات منصة لتحسين خدماتهم التجارية، وتحسين عملياتهم، واتخاذ قرارات مستنيرة قائمة على البيانات

الكلمات المفتاحية : الحوسبة السحابية ; إنترنت الأشياء ; التعلم الآلي ; الذكاء الاصطناعي