



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

MÉMOIRE DE MASTER RECHERCHE

En
Informatique

Option
Réseaux et Sécurité

Thème

L'automatisme intelligent de la collecte et le
stockage du sang via l'IoT

Présenté par : M^{lle} BELLABIOD Lydia
Et M^{lle} BAICHE Silia

Soutenu le 1^{er} Juillet 2024 devant le jury composé de :

Président	M ^{me} HOCINI	Kenza	MAB	U. A/Mira	Béjaia.
Examineur	M ^{lle} MAMMERI	Souhila	MCB	U. A/Mira	Béjaia.
Encadreurs	M. YAZID	Mohand	Professeur	U. A/Mira	Béjaia.
	M ^{me} BENLALA	Wissam	Doctorante	U. A/Mira	Béjaia.

✧ Remerciements ✧

En premier lieu, je remercie ALLAH, le Tout-Puissant, pour m'avoir accordé la force, la patience et la persévérance nécessaires pour mener à bien ce mémoire. C'est grâce à Sa miséricorde que j'ai pu accomplir ce travail.

Je suis particulièrement fière de l'engagement et de la détermination que j'ai apportés à ce projet. La finalisation de ce mémoire, en collaboration avec mon binôme "Lydia BELLABIOD", est une réussite dont je suis très satisfaite.

Je tiens à exprimer ma profonde gratitude à mon directeur de mémoire, "M. Mohand Yazid", pour son encadrement exceptionnel, ses précieux conseils et son soutien constant qui ont été déterminants dans la réalisation de ce mémoire.

Je remercie également Mme "Wissam BENCALAI" pour ses orientations précieuses durant les différentes phases de ce projet. Sa grande disponibilité tout au long de ce travail ainsi que ses remarques judicieuses et son expertise ont été d'une aide inestimable.

*Je tiens à exprimer ma sincère gratitude envers le jury de ce mémoire, présidé par le "Dr. Hocini KENZA" et l'examineur "Dr. Mammeri Souhila".
Leurs évaluations précieuses, leurs recommandations avisées et leur soutien durant cette étape sont essentiels pour perfectionner notre travail. Nous leur sommes reconnaissants pour leur dévouement et leur contribution indispensable à cette phase décisive de notre parcours académique.*

Mes plus sincères remerciements vont également à mes deux frères, mes parents, surtout à ma mère, pour leur soutien constant et leurs encouragements. Leur présence à mes côtés a été une véritable source de réconfort et de motivation tout au long de ce parcours.

✧ Remerciements ✧

Je tiens d'abord à exprimer ma profonde gratitude à Dieu, sans qui tout serait aléatoire dans ce parcours que j'ai suivi. Les mots ne suffisent pas pour exprimer ma reconnaissance, mais je ne peux dire que « Alhamdulillah » pour toutes les coïncidences et les bénédictions placées sur mon chemin.

Je suis particulièrement fière de l'engagement et de la détermination que j'ai apportés à ce projet. La finalisation de ce mémoire, en collaboration avec mon binôme "BAICHE Silia", est une réussite dont je suis très satisfaite.

Un merci très spécial à ma mère, pilier de ma vie, qui a sacrifié tant et combattu tous les obstacles pour me permettre de grandir et d'étudier. Mon amour pour toi est immense, ma source de vie.

Je remercie également Mme "Wissam BENLALA" pour ses orientations précieuses durant les différentes phases de ce projet. Sa grande disponibilité tout au long de ce travail ainsi que ses remarques judicieuses et son expertise ont été d'une aide inestimable.

Je tiens à exprimer ma sincère gratitude envers le jury de ce mémoire, présidé par le "Dr. Hocini Kenza" et l'examinatrice "Dr. Mammeri Souhila".

Leurs évaluations, leurs recommandations avisées et leur soutien durant cette étape sont essentiels pour perfectionner notre travail. Nous leur sommes reconnaissants pour leur dévouement et leur contribution indispensable à cette phase décisive de notre parcours académique.

Je remercie également mes sœurs et mon frère, toujours présents pour moi, soutenant chacun de mes pas. Votre présence est une source constante de réconfort et d'encouragement.

Je suis également reconnaissante envers ma meilleure amie et sœur, "Lydia OURLABAH". Ta présence à mes côtés a été un trésor. J'ai appris tant de toi et de ta sagesse, et pour cela, je te suis profondément reconnaissante.

À Lydia Ait.O, ma meilleure amie qui m'a accompagné tout au long de ces années, ton amitié est un cadeau précieux.

Enfin, je tiens à remercier infiniment ma binôme et amie, Silia Baiche, qui a été ma complice et mon soutien tout au long de mon parcours universitaire. Être à tes côtés a été non seulement un plaisir mais aussi une grande aide..

✧ *Dédicaces* ✧

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

Mon père "BAICHE Rachid" et surtout à mon trésor ma mère "OTMANI Fatima "

Ma petite soeur "BAICHE Manel"

Mon petit frère "BAICHE Islam"

ma cousine "AMRIOU Hanen"

à ma meilleure amie "Lydia BELLABIOD"

à moi-même "BAICHE Silia"

Table des matières

Table des matières	i
Table des figures	iv
Liste des tableaux	vi
Liste des équations	vii
Liste des algorithmes	viii
Liste des acronymes	ix
Introduction générale	1
1 Internet des objets	3
1.1 Introduction	3
1.2 Origines de l’IoT	3
1.3 Définition de l’Internet des Objets (IoT)	4
1.4 Composants essentiels de l’IoT	4
1.4.1 Objet IoT	4
1.4.2 Données	5
1.4.3 Connectivité	5
1.4.4 Cloud	5
1.5 Technologies IoT	5
1.5.1 Réseaux sans fil	6
1.5.2 Capteurs	7
1.5.3 Passerelles IoT	7
1.5.4 Protocoles de communication IoT	8
1.6 Les couches de l’IoT	8
1.6.1 Couche de perception	8
1.6.2 Couche de réseau	9
1.6.3 Couche de traitement	9
1.6.4 Couche d’application	9
1.7 Applications de l’IoT dans différents domaines	9

1.8	Conclusion	10
2	IoT dans secteur médical	11
2.1	Introduction	11
2.2	Définition de l’IoT dans le domaine médical	11
2.3	Composants essentiels de l’IoT dans le contexte médical	11
2.4	Les applications de l’IoT dans le domaine de la santé	12
2.5	Données médicales collectées par l’IoT	13
2.5.1	C’est quoi une donnée médicale?	13
2.5.2	Types de données médicales collectées par les appareils IoT	13
2.5.3	Technologies IoT pour la collecte de données médicales	14
2.6	Avantages potentiels de l’IoT pour le système de santé	17
2.7	Défis de la mise en œuvre de l’IoT dans les soins de santé	17
2.8	Le triptyque essentiel de la gestion et la fiabilité des données médicales : «Qualité, intégrité et confidentialité»	18
2.8.1	Mesures pour garantir la qualité et l’intégrité des données :	18
2.8.2	Sécurité et confidentialité des données médicales	18
2.9	Revue des travaux connexes	18
2.10	Analyse et Discussion	31
2.11	Conclusion	31
3	Plateforme IoT pour le stockage et la gestion automatisée du de sang	32
3.1	Introduction	32
3.2	Système automatisé de gestion du sang : une nécessité pour des soins de santé modernes	32
3.3	Aperçu global de l’architecture proposée	33
3.4	Justification de choix de chaque composant intégré	35
3.4.1	Capteur de pression (Honeywell 26PC)	35
3.4.2	Capteur d’humidité et de la température (SI7021)	36
3.4.3	Capteur de tension électrique (ZMPT101B)	36
3.4.4	Microcontrôleur ESP32	37
3.4.5	Fog Computing (Raspberry Pi)	37
3.4.6	Passrelle (LoRAWAN)	38
3.4.7	Cloud	38
3.4.8	Twilio	39
3.4.9	Firebase	39
3.5	Fonctionnement de l’architecture	40
3.5.1	Collecte des données	40
3.5.2	Transmission vers le Fog Computing (Raspberry Pi)	43
3.5.3	Traitement des données par le Fog Computing	43
3.5.4	Transmission vers le Cloud	44

3.6	Algorithme globale de gestion du stockage de sang	45
3.6.1	Explication de l'algorithme	48
3.7	Partie pratique et simulation	52
3.7.1	Environnement de programmation (Google Colab)	52
3.7.2	Algorithme Python globale de surveillance et de gestion des stocks de sang	52
3.7.3	Déscription de programme python	56
3.7.4	Exécution de l'algorithme Python dans Google Colab	59
3.8	Carte de fidélité de sang - Avantages et fonctionnalités	65
3.8.1	Avantages de la carte	65
3.8.2	Caractéristiques de la carte	65
3.9	Interface de don de sang connectée	66
3.9.1	Objectifs principaux de la plateforme web	66
3.9.2	Firebase : services principaux utilisés	66
3.9.3	Présentation des pages de l'interface Web	67
3.9.4	Gestion des utilisateurs par le manager	72
3.10	Conclusion	73
3.11	Travaux futurs	74
	Conclusion et perspectives	75
	Bibliographie	77

Table des figures

1.1	Internet des objets[1].	3
1.2	Une architecture générale des services IoT [101].	5
1.3	Les différents appareils et réseaux[111].	7
1.4	Couches IoT	9
2.1	Tableau de bord OCARIoT dans l'architecture globale [82].	19
2.2	Approche de conception [117].	20
2.3	La topologie et l'architecture du framework IoMT proposé[131].	21
2.4	Architecture Diabloop[105].	22
2.5	Algorithme de base de Diabloop[105].	22
2.6	L'architecture EcoHealth[104].	23
2.7	Les technologies utilisées pour implémenter les modules EcoHealth[104]	23
2.8	Classification des crises épileptiques basée sur les canaux multiples (MCSV)[130].	24
2.9	Architecture système de la classification proposée des crises épileptiques[130].	24
2.10	Architecture générale multi-couche[87].	25
2.11	L'interface utilisateur[87].	25
3.1	Chambre automatisée de stockage de sang en 3D.	34
3.2	Diagramme représentatif de l'architecture pour la gestion automatisée du stockage du sang.	35
3.3	Capteur de pression (Honeywell 26PC)[34].	36
3.4	Capteur SI7021)[35].	36
3.5	Capteur de tension électrique (ZMPT101B) [78].	37
3.6	Microcontrôleur ESP32 [34].	37
3.7	Raspberry Pi[23].	38
3.8	Importation des bibliothèques.	57
3.9	Définition des variables globales.	57
3.10	Génération des données aléatoires.	58
3.11	Mesure du temps d'exécution.	58
3.12	Conversion en DataFrame Pandas.	58
3.13	Affichage des données et statistiques.	58
3.14	Tracé des donnée.	59
3.15	Exécution de la fonction de tracé.	59
3.16	Dataset utilisé.	60

3.17	Notifications envoyées.	61
3.18	Temps d'exécution.	61
3.19	Diagrammes à barres illustrant la quantité de chaque type de sang[79].	62
3.20	Évolution temporelle de la quantité de chaque type de sang en courbes[79].	63
3.21	Présentation sous forme de diagramme circulaire des notifications reçues par le manager[79].	63
3.22	Diagramme circulaire représentant les notifications normales vers le cloud selon le type de sang[79].	64
3.23	Diagramme circulaire représentant les notifications urgentes vers le cloud selon le type de sang[79].	64
3.24	Page d'accueil.	67
3.25	Page inscription.	68
3.26	Page de connexion.	69
3.27	Tableau de board des donneurs.	70
3.28	Page de contact.	71
3.29	Données des donneurs sur Firestore	72
3.30	Commentaires envoyés par les donneurs via la page de contact.	72

Liste des tableaux

- 2.1 Capteurs IOMT 16
- 2.2 Comparaison des articles IoT pour la santé. 26
- 3.1 Tableau récapitulatif des principaux critères de l'architecture proposée. 73

Liste des équations

Équation 3.1 : Quantité de sang en fonction de la pression

Équation 3.2 : Calcul du pourcentage de sang dans chaque case .

Équation 3.3 : Exemple de calcul du pourcentage de sang.

Liste des algorithmes

1	Surveillance du stockage de sang	44
2	Système de surveillance et gestion du stockage de sang	46
3	La suite de l’algorithme système de surveillance et gestion du stockage de sang	47
4	Initialisation des types de sang et des seuils critiques	48
5	Données de la chambre de stockage	48
6	Initialisation des capteurs et lecture des données	49
7	Vérification des conditions et activation de l’UPS	49
8	Lire les données de capteur de pression dans chaque case de chaque étage . .	50
9	Connexion du microcontrôleur et transmission des données	50
10	Traitement des Données par le Fog et Transmission au Cloud	51
11	Interface utilisateur pour les notifications	51
12	Fonction d’allumage de la LED critique	51

Liste des acronymes

ALPHABET	ACRONYME	Signification
A	1. API 2. ADC 3. ADN 4. ANN	1. Application Programming Interface 2. Analog-to-Digital Converter 3. Acide désoxyribonucléique 4. Artificial Neural Networks
C	1. COV	1. Composés organiques volatils
D	1. DSS	1. Decision Support System
E	1. ECG 2. EEG 3. EMG 4. ESP32	1. Électrocardiogramme 2. Électroencéphalogramme 3. Électromyogramme 4. Espressif Systems
G	1. GPRS	1. General Packet Radio Service
H	1. HIPAA 2. HRV	1. Health Insurance Portability and Accountability Act 2. Heart Rate Variability
I	1. IoT 2. IoMT 3. IMC	1. Internet des Objets 2. Internet of Medical Things 3. Indice de Masse Corporelle
L	1. LoRaWAN	1. Long Range Wide Area Network
M	1. MQTT	1. Message Queuing Telemetry Transport
O	1. OCARIoT	1. Open Consumption and Application of Real-time IoT data
R	1. RGPD	1. Règlement Général sur la Protection des Données
S	1. SVM 2. SI7021	1. Support Vector Machines 2. Silicon Labs 7021
U	1. UML 2. UPS	1. Unified Modeling Language 2. Uninterruptible Power Supply
W	1. WBAN	1. Wireless Body Area Network

Introduction générale

L'Internet des Objets (IoT) représente une révolution technologique majeure qui transforme notre environnement numérique en connectant et en faisant communiquer les appareils de manière transparente. Grâce à des innovations telles que les smartphones, le Cloud computing et les capteurs intelligents, l'IoT permet la collecte et le traitement automatisés des données en temps réel, sans intervention humaine. Il repose sur une architecture multicouche sophistiquée qui gère le parcours des données, de leur acquisition initiale à leur analyse via des plateformes Cloud. Cette infrastructure dépasse largement les simples applications technologiques pour englober une multitude de secteurs économiques et sociaux, notamment le domaine de la santé, en facilitant la personnalisation des soins, l'optimisation industrielle et le développement de villes intelligentes. Elle promet une gestion optimisée des ressources et une automatisation accrue des processus.

Dans ce mémoire nous nous penchons sur l'application de l'Internet des Objets (IoT) dans le domaine médical, en particulier sur la gestion critique du sang, qui est un défi majeur.

Chaque année, la disponibilité de sang sûr et de qualité est essentielle pour sauver des millions de vies. Cependant, les systèmes actuels de gestion du sang se révèlent souvent inefficaces, entraînant des gaspillages, des pénuries et mettant en danger la santé des patients. Pour relever ces défis, nous proposons une architecture IoT innovante basée sur le fog/cloud computing pour le stockage et la gestion automatisée du sang. Cette architecture s'appuie sur des technologies de connectivité telles que le Wi-Fi et le Bluetooth, ainsi que sur des protocoles de communication comme MQTT et LoRa, afin d'assurer une collecte et une transmission en temps réel des données critiques sur le sang. En exploitant ces données, notre système IoT permettra d'optimiser les niveaux de stock, de surveiller l'état du sang et d'automatiser les processus de distribution, garantissant ainsi une gestion du sang plus efficace, plus sûre et plus responsable.

Pour une meilleure appréhension de l'IoT et de ses applications, ainsi que pour comprendre l'architecture IoT que nous proposons, nous recommandons de suivre Cette méthodologie de présentation de ce mémoire comprend trois chapitres structurés :

- Le premier chapitre aborde l'internet des Objets (IoT) de manière générale, en présentant sa définition, son historique, ses composants, ainsi que son fonctionnement et ses applications dans divers domaines.
- Le deuxième chapitre se concentre sur l'IoT dans le domaine médical, en étudiant ses applications, son importance et son architecture. Ce chapitre inclut également une étude approfondie d'un ensemble d'articles sélectionnés pour explorer en détail l'internet des Objets médical.
- Dans le troisième chapitre, nous présentons notre proposition d'exploiter la technologie IoT pour apporter des améliorations dans le domaine médical. Nous proposons et décrivons une plateforme IoT pour la gestion et le stockage automatisé du sang, en détaillant son architecture, son importance et ses composants.

Ce mémoire se conclura par une conclusion générale et des perspectives de recherche découlant de notre travail.

Internet des objets

1.1 Introduction

Dans cette section, nous découvrirons les fondements de l'IoT, ses principaux enjeux, ainsi que son application générale, illustrant comment cette technologie façonne divers domaines, y compris celui des soins de santé.

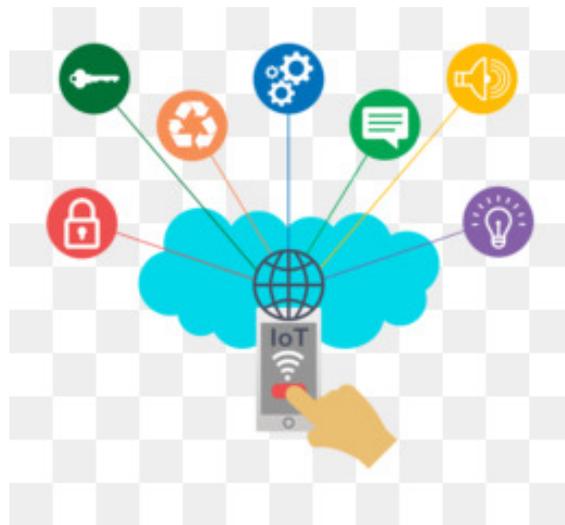


FIGURE 1.1 – Internet des objets[1].

1.2 Origines de l'IoT

Les origines de l'Internet des Objets (IoT) remontent aux simples appareils interconnectés en 1960. C'est à cette époque que des chercheurs ont exploré les concepts de réseau informatique, et dans les années 1980 ont vu l'émergence des réseaux locaux (LAN) permettant la communication en temps réel entre ordinateurs personnels. Ensuite, dans les années 1990, Le développement du RFID et des réseaux sans fil facilite la connectivité des objets. Alors en 1999, Kevin Ashton a inventé le terme "Internet des Objets", en visionnant un monde où les objets physiques se connecteraient grâce à l'Identification par Radiofréquence (RFID) et aux micro-puces. Cette vision a été concrétisée par les progrès technologiques

des décennies suivantes, tels que les smart-phones, le cloud computing, l'amélioration des processeurs et des logiciels, qui ont permis une gestion robuste des données. ainsi Le développement de capteurs sophistiqués a permis de surveiller diverses conditions (mouvement, température, taux d'humidité, etc.). Aujourd'hui, l'adoption généralisée des appareils mobiles et de la connectivité sans fil a créé un réseau quasi-omniprésent reliant les personnes et les objets, avec plus de 7 milliards d'appareils IoT connectés à l'heure actuelle, les prévisions des experts indiquent une croissance significative, atteignant potentiellement 10 milliards d'ici 2020 et 22 milliards d'ici 2025[2].

1.3 Définition de l'Internet des Objets (IoT)

L'Internet des objets (IoT) est un réseau de dispositifs physiques interconnectés qui peuvent transférer des données et se communiquer entre eux [108] sans nécessiter d'intervention humaine.

Un objet IoT doit principalement être [95] :

- Intelligent : L'objet doit être doté d'une capacité de traitement de l'information lui permettant de collecter, stocker et manipuler des données. Il peut ainsi effectuer des calculs.
- Autonome : L'objet doit pouvoir fonctionner de manière indépendante, sans nécessiter d'intervention humaine
- L'objet doit être capable de se connecter à d'autres objets ou à un réseau, généralement via des technologies de communication sans fil (Wi-Fi, Bluetooth, Zigbee, etc.) ou filaires (Ethernet).

1.4 Composants essentiels de l'IoT

Les quatre piliers de l'IoT sont des composants essentiels qui forment la base de fonctionnement des systèmes basés sur l'IoT, permettant aux dispositifs interconnectés de générer et d'utiliser des données efficacement. Voici les points les plus importants pour chaque composant :

1.4.1 Objet IoT

Cet appareil fait référence aux objets physiques (tels que les téléphones, les voitures et les appareils électroménagers) qui sont connectés à l'internet[93] [3]. Ces objets connectés sont des dispositifs équipés de capteurs, Ils sont ainsi capables de collecter des données, de les traiter, de prendre des décisions autonomes en fonction de ces données, et ils peuvent également interagir avec d'autres objets connectés[110].

1.4.2 Données

Sont les informations recueillies par les appareils IoT. Ce pilier se concentre sur l'exploitation de ces données pour améliorer les performances du réseau, prendre des décisions éclairées et améliorer la fonctionnalité globale du système.[93]

1.4.3 Connectivité

Ce composant [93] assure qu'il y a une transmission fluide des données entre les dispositifs et les autres composants du système IoT (cloud computing). L'importance de cette fonction augmente avec le besoin de traitement des données en temps réel, où toute interruption de la connexion internet est inacceptable

1.4.4 Cloud

L'analyse implique l'examen et l'interprétation des données collectées. Cela peut se produire directement sur l'appareil, passerelles, ou dans le cloud [93]. L'analyse des données aide à dériver des informations significatives à partir des données collectées, ce qui peut influencer les processus de prise de décision et optimiser les performances[3].

1.5 Technologies IoT

L'Internet des objets (IoT) englobe un large éventail de technologies qui permettent aux objets physiques de se connecter et d'échanger des données entre eux. Cela inclut un large éventail de dispositifs, de capteurs aux réseaux sans fil et aux plateformes de traitement des données[84].

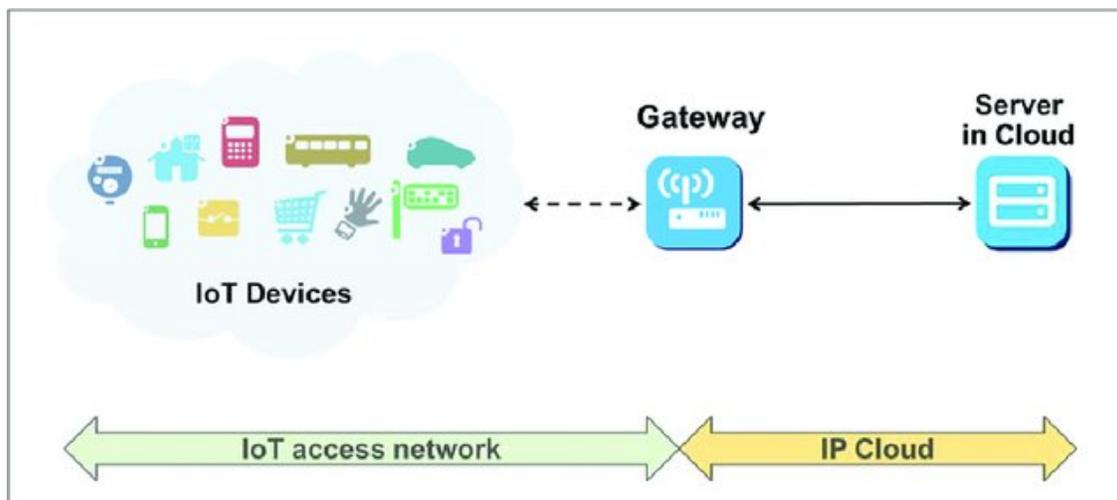


FIGURE 1.2 – Une architecture générale des services IoT [101].

1.5.1 Réseaux sans fil

Parmi les technologies de réseau sans fil les plus couramment utilisées pour l'IoT :

- Wi-Fi : Le Wi-Fi [100] permet des taux de transfert de données locaux très élevés nécessitant une bande passante élevée et résultant une consommation élevée. Il est généralement déployé sur des dispositifs tels que les ordinateurs portables et les smartphones.
- Bluetooth : Le Bluetooth[81] permet de connecter sans fil divers appareils tels que les kits mains libres, les smartphones et les ordinateurs sur de courtes distances. La portée varie en fonction de la classe de l'appareil : jusqu'à 20 mètres pour la classe 2 et jusqu'à 100 mètres pour la classe 1.
- Bluetooth Low Energy : est une technologie conçue pour une faible consommation d'énergie et un faible coût[94], tout en maintenant une portée de communication similaire au Bluetooth classique[112]. Il est couramment utilisé dans les appareils IoT[132]. L'intégration du mode basse consommation (BLE) dans le Bluetooth 4.0 a propulsé le développement des objets connectés[92].
- Zigbee : Zigbee [91] est un standard de communication sans fil à Faible consommation d'énergie pour une longue durée de vie des batteries, avec Débit de données allant jusqu'à 250 kbit/s. Il est utilisé dans de nombreuses applications telles que la domotique, l'éclairage connecté, la surveillance, le contrôle industrie, et aussi dans l'IoT.
- Z-Wave : Z-wave est une alternative à Zigbee pour la domotique[91]. La différence entre Z-Wave et Zigbee réside dans la fréquence utilisée : Z-Wave fonctionne sur une fréquence inférieure à 1 GHz, ce qui est plus bas que la fréquence de 2,4 GHz utilisée par Zigbee [4].
- LPWAN : Les réseaux LPWAN (Low-Power Wide Area Networks) [85] sont idéaux pour l'Internet des objets (IoT). L'IoT est massivement utilisé dans les transports, l'énergie, les villes intelligentes, l'électronique, l'industrie, l'environnement et l'agriculture. Ils sont conçus pour connecter des appareils sur de longues distances avec une consommation d'énergie minimale. Parmi les exemples de LPWAN : LoRa et Sigfox.

Vertical Industries

Different devices and network

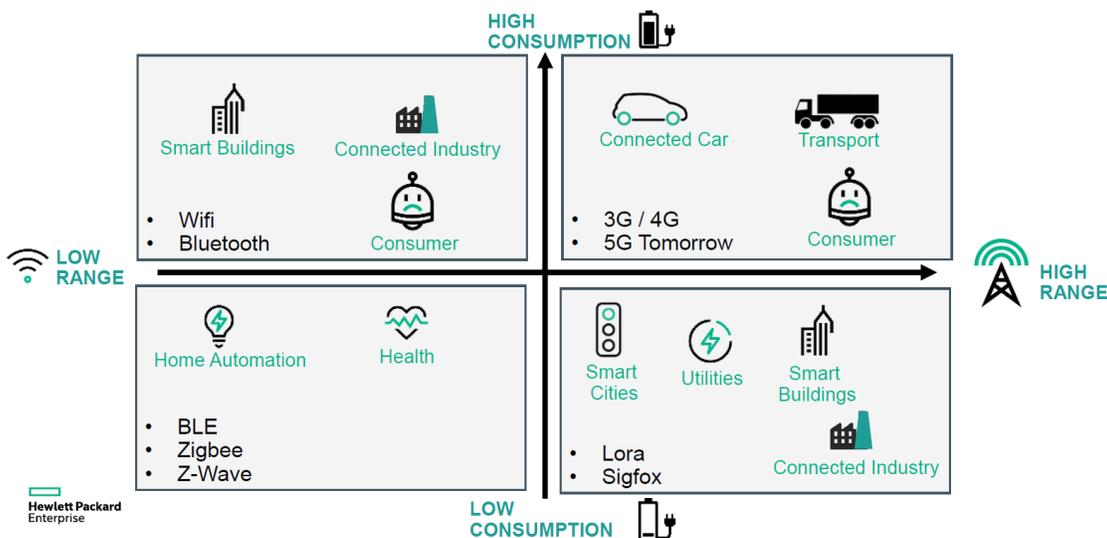


FIGURE 1.3 – Les différents appareils et réseaux[111].

1.5.2 Capteurs

Les capteurs sont des appareils qui peuvent surveiller l'environnement dans de nombreuses applications. Ils sont programmés pour recueillir des données et les transmettre vers un nœud de passerelle, un module de traitement de données ou un nœud récepteur[112]. Parmi les exemples de capteurs, on trouve :

- Capteurs environnementaux : Mesurent des conditions telles que la température, l'humidité, la pression, la lumière et la qualité de l'air.
- Capteurs de mouvement : Détectent le mouvement, la vibration et l'orientation.
- Capteurs biométriques : Suivent les données physiologiques telles que la fréquence cardiaque, la pression artérielle et le niveau de glucose dans le sang.
- Capteurs d'image et vidéo : Capturent des images et des vidéos pour la surveillance, la reconnaissance d'objets et l'analyse visuelle.
- Capteurs de proximité : Détectent la présence d'objets à proximité.

1.5.3 Passerelles IoT

Une passerelle d'IoT est un appareil matériel ou un logiciel qui agit comme un pont entre les appareils IoT utilisant différents protocoles de communication (Bluetooth, Zigbee, Wi-Fi, LoraWAN, etc.) et le cloud (ou d'autres systèmes). Elles traduisent les données de divers protocoles IoT en formats standardisés (souvent basés sur IP), compatibles avec

les systèmes cloud.[6] Les passerelles peuvent effectuer un pré-traitement des données des capteurs, réduisant ainsi la quantité de données envoyées au cloud, optimisant l'utilisation de la bande passante et améliorant l'autonomie des appareils alimentés par batterie[5].

Elles peuvent aussi mettre en œuvre des mécanismes de sécurité [5] comme le cryptage, l'authentification et la gestion des autorisations pour protéger les données IoT pendant le transit et le stockage.

1.5.4 Protocoles de communication IoT

Ces protocoles assurent que les dispositifs sont capables de se localiser et de s'interconnecter, de transférer des informations de façon sécurisée et performante.

- Message Queuing Telemetry Transport (MQTT) : Un protocole de publication/abonnement léger idéal pour connecter des dispositifs distants avec une bande passante réseau minimale[46].
- Constrained Application Protocol (CoAP) : Un protocole RESTful conçu pour les appareils à ressources limitées, permettant une communication efficace sur des réseaux à faible puissance[49].
- Extensible Messaging and Presence Protocol (XMPP) : Utilisé pour la messagerie instantanée et la détection de présence en temps réel dans l'IoT, en exploitant une architecture flexible et décentralisée[48].
- Hypertext Transfer Protocol Secure (HTTPS) : Une extension de HTTP, elle assure une communication sécurisée en chiffrant les données à l'aide de TLS/SSL, les protégeant des espions et garantissant l'intégrité des données et l'authentification entre le client et le serveur.[47].

1.6 Les couches de l'IoT

L'architecture de l'Internet des Objets (IoT) est généralement structurée en plusieurs couches, Voici une description des couches principales qui constituent l'architecture typique de l'IoT [93] :

1.6.1 Couche de perception

Cette couche est la plus bas niveau de l'architecture d'IoT. Elle est constituée de capteurs et de dispositifs intégrés dans l'architecture de l'IoT, utilisés pour recueillir des données environnementales.

1.6.2 Couche de réseau

La couche de réseau est responsable de la transmission des données collectées par les capteurs à d'autres dispositifs ou à l'internet. Elle utilise diverses technologies de communication, telles que Bluetooth, Zigbee, Wi-Fi, 4G/5G, LPWANs.

1.6.3 Couche de traitement

Cette couche gère les données reçues, les traite et les stocke de manière appropriée. Cette couche peut également inclure des fonctionnalités de traitement des données, comme l'analyse de données, l'apprentissage automatique et la gestion de base de données, cela inclut le cloud computing.

1.6.4 Couche d'application

La couche d'application est celle qui présente les données traitées aux utilisateurs finaux sous une forme utilisable. Cela inclut les applications mobiles, l'interface du site web et les assistants vocaux tels qu'Amazon Alexa ou Google Home.

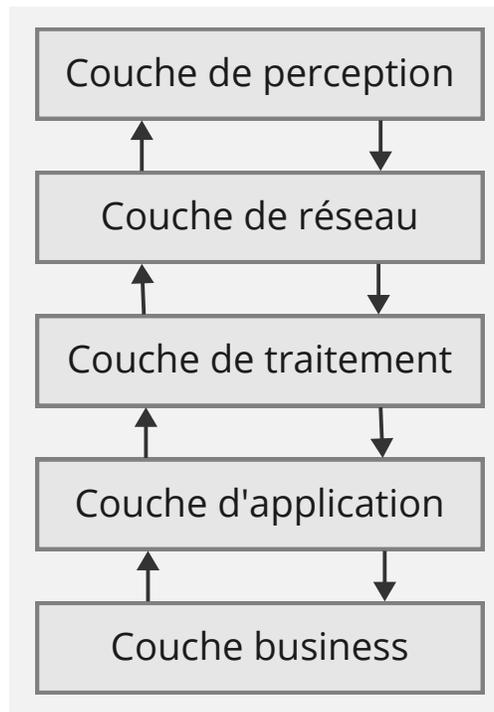


FIGURE 1.4 – Couches IoT

1.7 Applications de l'IoT dans différents domaines

L'Internet des Objets (IoT) révolutionne de nombreux secteurs grâce à sa capacité à connecter et à automatiser des processus dans divers domaines.

- ▷ Dans l'industrie, l'IoT permet une surveillance des machines, offrant des possibilités de maintenance prédictive pour prévenir les pannes et optimiser la production. L'automatisation par des robots et des systèmes autonomes améliore la productivité et la sécurité, tandis que la gestion avancée des stocks l'efficacité, réduit les gaspillages et assure une meilleure traçabilité des produits. [38]
- ▷ Le secteur de la santé bénéficie également de l'IoT, notamment pour le suivi à distance des patients, permettant une surveillance continue des signes vitaux et une intervention rapide en cas d'anomalies. Les technologies IoT supportent également le diagnostic et le traitement des maladies avec plus de précision, et optimisent la gestion des hôpitaux en améliorant le suivi des flux de patients et la gestion des ressources. [120]
- ▷ En agriculture, l'IoT transforme les pratiques traditionnelles avec des systèmes d'irrigation intelligente qui ajustent l'eau en fonction des besoins réels des cultures, améliorant ainsi la conservation de l'eau et les rendements agricoles[116]. La surveillance des cultures via des capteurs IoT aide à détecter rapidement les maladies ou les infestations, et la traçabilité des produits alimentaires renforce la sécurité alimentaire[126].
- ▷ Dans d'autres domaines tels que les 'villes intelligentes' et les 'maisons intelligentes', l'IoT améliore la qualité de vie en optimisant la gestion du trafic, l'éclairage public, et en automatisant les appareils domestiques pour plus de confort et de sécurité [98]. Enfin, dans le domaine de l'environnement, l'IoT aide à surveiller la pollution et à gérer les ressources naturelles, contribuant à la protection de l'environnement et à la lutte contre le changement climatique[39].

1.8 Conclusion

En somme, ce chapitre introductif a fourni une analyse des bases de l'Internet des Objets (IoT), décrivant ses composantes clés et explorant ses multiples applications. Cette compréhension initiale sera approfondie dans le chapitre suivant, où nous tenterons d'analyser comment l'IoT est spécifiquement appliqué et intégré dans le secteur médical.

IoT dans secteur médical

2.1 Introduction

L'adoption de l'Internet des Objets (IoT) dans le domaine médical a été au centre des préoccupations ces dernières années, témoignant d'un fort engagement envers l'innovation. Cette évolution représente une transformation radicale du paysage médical, stimulant des recherches visant à optimiser la gestion des patients, des données médicales et des traitements. En effet, l'intégration de l'IoT offre une pléthore d'avantages, notamment l'amélioration des systèmes intelligents pour le suivi des maladies chroniques et une optimisation globale des pratiques médicales.

Dans ce cadre, la collecte, l'analyse et le traitement des données sont essentiels. Comment ces processus se déroulent-ils alors dans l'utilisation de l'IoT dans le domaine médical ?

2.2 Définition de l'IoT dans le domaine médical

L'Internet des objets (IoT) dans le domaine de la santé désigne l'intégration de dispositifs intelligents équipés de capteurs spécifiques et d'autres technologies IoT dans les processus et les objectifs médicaux en collectant, transmettant et analysant des données.[7] Cette intégration vise à améliorer les soins aux patients, à accroître l'efficacité opérationnelle et à faciliter la surveillance et la gestion en temps réel des conditions de santé.[8]

2.3 Composants essentiels de l'IoT dans le contexte médical

Un système IoT médical efficace est constitué de plusieurs composants fondamentaux :

- Objets connectés : Il s'agit de dispositifs tels que des capteurs biométriques, des appareils portables et des équipements de télésanté, qui sont responsables de la collecte et de la transmission des données médicales.

- Connectivité : Les réseaux sans fil, incluant le Wi-Fi, le Bluetooth et les réseaux cellulaires, permettant la communication entre les objets connectés et les systèmes de gestion des données.
- Plateformes de gestion et d'analyse des données : infrastructure informatique pour stocker, traiter, analyser et visualiser les données médicales collectées.
- Interfaces utilisateur et applications mobiles : permettent aux patients, aux professionnels de santé et aux chercheurs d'accéder aux données médicales et d'interagir avec le système.

2.4 Les applications de l'IoT dans le domaine de la santé

L'Internet des Objets (IoT) transforme la médecine connectée de plusieurs façons :

- Surveillance de santé à distance : Les objets connectés (IoT) peuvent suivre en continu les signes vitaux des patients (comme la tension artérielle, le rythme cardiaque et la glycémie) et transmettre ces données en temps réel aux professionnels de santé. [99] Cela permet une détection précoce des problèmes, des soins personnalisés et une réduction des hospitalisations.[9]
- Gestion des médicaments : Les piluliers intelligents et les distributeurs de médicaments peuvent rappeler aux patients de prendre leurs médicaments à l'heure, suivre l'observance du traitement et alerter les aidants ou les professionnels de santé en cas d'oubli de dose.[115]
- Aide aux personnes âgées : Les dispositifs IoT peuvent surveiller les activités des personnes âgées, détecter les chutes et leur rappeler les tâches quotidiennes, améliorant ainsi leur sécurité et leur autonomie à domicile.[109]
- Gestion des maladies chroniques : Les dispositifs IoT peuvent aider les personnes atteintes de maladies chroniques comme le diabète ou l'asthme à gérer leur santé en surveillant les symptômes, en suivant la prise de médicaments et en fournissant un soutien et des conseils personnalisés.[88]
- Hôpitaux intelligents : Les objets connectés peuvent être utilisés pour suivre les équipements médicaux, surveiller les conditions environnementales et optimiser les flux de travail hospitaliers [125], améliorant l'efficacité et les soins aux patients.
- Intervention d'urgence : Les dispositifs connectés peuvent détecter rapidement les urgences telles que les chutes ou les crises cardiaques et alerter automatiquement les services d'urgence [89], ce qui peut sauver des vies.

2.5 Données médicales collectées par l'IoT

2.5.1 C'est quoi une donnée médicale ?

Les données médicales englobent toutes les informations relatives à la santé et à l'historique médical d'un individu ou populations.[10] Les données se présentent sous diverses formes, notamment : nombres, texte, données codées, graphiques, et sons.[97]

Elles comprennent des éléments tels que les diagnostics, les traitements suivis, les médicaments prescrits, les résultats de tests, la facturation, la production de rapports médicaux [97], et autres informations essentielles à la gestion de la santé.

2.5.2 Types de données médicales collectées par les appareils IoT

Les dispositifs IoT dans le domaine médical recueillent divers types de données qui peuvent être classifiés en plusieurs catégories principales. Chaque catégorie joue un rôle essentiel dans l'analyse de la santé du patient et la prise de décisions médicales.

2.5.2.1 Données physiologiques

Les données physiologiques [118] permettent aux professionnels de santé de détecter des anomalies physiologiques qui pourraient indiquer des conditions médicales aiguës ou chroniques, facilitant ainsi des interventions précoces et personnalisées.

Ces données concernent les fonctions corporelles vitales et incluent :

- Fréquence cardiaque (HRV)
- Pression artérielle
- Température corporelle
- Saturation en oxygène (SpO2)
- Électromyogramme (EMG) : Évalue l'activité musculaire.

2.5.2.2 Données comportementales

Ces informations comportementales [123] portent sur les habitudes de vie et les comportements quotidiens, souvent collectées via des capteurs et des questionnaires. En analysant ces données, les médecins peuvent recommander des modifications comportementales qui améliorent la santé globale et préviennent des maladies liées au mode de vie, comme les maladies cardio-vasculaires. Parmi ces informations, on a :

- L'activité physique : Nombre de pas, distance parcourue, calories brûlées.

- Le sommeil : Durée, qualité, cycles de sommeil.
- L'alimentation : Apport calorique.
- Humeur et qualité de vie : Évaluées par des questionnaires sur la dépression et l'anxiété.

2.5.2.3 Données environnementales

Données environnementales[118][121] aident à établir des liens entre les conditions environnementales et les problèmes de santé, permettant de prendre des mesures pour minimiser l'exposition à des environnements nocifs. On peut citer :

- Température et humidité
- Niveau de luminosité

2.5.2.4 Données génétiques et biométriques

Les données génétiques et biométriques[37] fournissent des informations sur le patrimoine génétique et les caractéristiques biologiques uniques d'un individu comme empreintes digitales, Séquence d'ADN, Mutations génétiques. Elles sont importantes pour la médecine personnalisée, permettant des traitements sur mesure basés sur le profil génétique du patient pour maximiser l'efficacité et minimiser les effets secondaires des médicaments.

2.5.2.5 Données cliniques et d'imagerie médicale

Incluent les résultats d'analyses médicales, radiographies, IRM, et autres types d'images diagnostiques qui sont fondamentales pour diagnostiquer diverses conditions médicales.

2.5.3 Technologies IoT pour la collecte de données médicales

1. **Montres intelligentes et bracelets d'activité** : Suivent divers paramètres de santé et sont populaires pour leur facilité d'utilisation et leur intégration avec les smartphones (Apple Watch, Fitbit, Samsung Galaxy Watch, etc.).[40]
2. **Patches cutanés** : offrent un suivi discret pour des conditions de santé spécifiques. Par exemple, des systèmes de surveillance continue du glucose utilisés par les personnes diabétiques[41], comme le Dexcom G6 et le Freestyle Libre[42].
3. **Vêtements intelligents** : Intègrent des capteurs pour un suivi sans entrave, adaptés à un usage quotidien (T-shirts, brassières, etc.)[43].
4. **Appareils de télésanté et systèmes de surveillance à distance**
Il existe plusieurs types d'appareils et leurs fonctionnalités

- **Appareils connectés** : Stéthoscopes numériques, Tensiomètres connectés, Thermomètres intelligents, Spiromètres. Ces dispositifs permettent aux professionnels de santé de surveiller à distance les signes vitaux et les conditions de santé des patients[44].

La télésanté permet une surveillance continue et réduit le besoin de visites hospitalières, ce qui est particulièrement avantageux pour les populations rurales ou isolées[45].

5. Systèmes de maison intelligente et solutions domotiques

Il existe plusieurs systèmes et capteurs, voici quelque types des capteurs utilisés [51] :

- Capteurs de température : Mesurent les variations de température dans l'environnement domestique.
- Capteurs d'humidité : Enregistrent les niveaux d'humidité dans l'air. Une humidité correctement régulée peut prévenir les problèmes de santé respiratoires et améliorer le confort général.
- Capteurs de qualité de l'air :
 - Détecteurs de CO₂ : Mesurent la concentration de dioxyde de carbone dans l'air, ce qui est important pour s'assurer que l'environnement intérieur n'est pas excessivement chargé en CO₂, ce qui peut affecter la santé et la cognition.
 - Détecteurs de CO : Détectent le monoxyde de carbone, un gaz toxique incolore et inodore, souvent produit par des appareils de chauffage défectueux ou mal ventilés.
 - Détecteurs de particules fines (PM_{2.5} et PM₁₀) : Mesurent la concentration de particules fines dans l'air, qui peuvent provenir de la pollution extérieure ou de certaines activités intérieures. Ces particules sont connues pour leurs effets néfastes sur la santé respiratoire et cardiovasculaire.
 - Détecteurs de composés organiques volatils (COV) : Surveillent les niveaux de divers composés organiques qui peuvent émaner de peintures, solvants, produits de nettoyage, et autres sources, et qui peuvent affecter la qualité de l'air intérieur.
 - Capteurs de lumière : Détectent les niveaux d'éclairage, permettant d'ajuster automatiquement l'éclairage pour un confort optimal et pour économiser de l'énergie.
 - Détecteurs de fumée et de feu : Essentiels pour la sécurité, ils alertent les occupants en cas de détection de fumée ou de feu dans la maison.

6. Exemples de capteurs et leurs spécificités

TABLEAU 2.1 – Capteurs IOMT

Nom du Capteur	Utilisation	Année de Fabrication
Thermistors[59]	Mesure de la température corporelle	1930
Polar H7[61]	Surveillance cardiaque	2013
Omron BP786N[62]	Surveillance de la pression artérielle	2014
Dexcom G6[63]	Surveillance du glucose en continu	2018
Masimo MightySat[64]	Oxymétrie de pouls	2015
ResMed AirSense 10[65]	Suivi du sommeil et troubles respiratoires	2014
Fitbit Charge[66]	Surveillance de divers paramètres physiologiques	2014
Apple Watch Series[60]	Surveillance cardiaque, détection de chute	2015
Medtronic Guardian Sensor 3[67]	Surveillance du glucose en continu	2017
Philips Respironics DreamStation[68]	Traitement des troubles du sommeil	2015
Withings Thermo[69]	Mesure de température sans contact	2016
AliveCor KardiaMobile[70]	Réalisation d'ECG à domicile	2012
GE Healthcare CARESCAPE[71]	Surveillance continue de plusieurs paramètres physiologiques	2023
Abbott FreeStyle Libre[72]	Surveillance du glucose sans prélèvements sanguins	2014
Abbott FreeStyle Libre 2[72]	Surveillance du glucose avec alertes	2020
Dexcom G7[75]	Surveillance du glucose en continu, améliorée	2022
BioButton[77]	Surveillance continue de plusieurs paramètres de santé	2021
Withings ScanWatch[74]	Détecte l'apnée du sommeil et autres, réalise un ECG,	2020
Philips Biosensor BX100[76]	Surveillance à distance des patients hospitalisés	2020
Masimo SafetyNet[64]	Surveillance à distance pour patients COVID-19	2020
Apple Watch Series 7[60]	Surveillance de santé avancée, incluant l'oxygène sanguin	2021
Oura Ring Generation 3[73]	Surveillance de la santé via une bague intelligente	2021

2.6 Avantages potentiels de l'IoT pour le système de santé

L'Internet des Objets (IoT) offre des avantages significatifs dans le secteur médical, touchant aussi bien les médecins, les hôpitaux que les patients :

- Pour les médecins : L'IoT renforce la capacité des médecins à être plus proactifs et attentifs. Les données recueillies par divers appareils connectés leur permettent de personnaliser le traitement des patients et d'atteindre des résultats optimaux. [53][107]
- Pour les hôpitaux : L'IoT améliore la capacité des hôpitaux à surveiller continuellement l'état de santé des patients grâce à une variété d'appareils[113]. Il facilite également le suivi en temps réel du positionnement d'équipements médicaux essentiels tels que les fauteuils roulants et les pompes à oxygène, grâce à des capteurs IoT [55].
- Pour les patients : Les capteurs portables et les dispositifs IoT jouent un rôle déterminant dans le suivi régulier de la santé et le diagnostic des maladies. Ils permettent aux professionnels de la santé d'analyser les données de santé de manière approfondie et réactive.[54][107]

2.7 Défis de la mise en œuvre de l'IoT dans les soins de santé

Malgré son potentiel considérable, l'intégration de l'IoT dans les soins de santé se heurte à de nombreux obstacles :

- La protection des données de santé sensibles des patients est une priorité absolue pour se protéger contre les violations de données et les accès non autorisés.[114]
- Il est essentiel d'assurer un échange de données transparent et une compatibilité entre les divers dispositifs IoT et les systèmes de santé pour une gestion et une utilisation efficaces des données.[107]
- Les établissements de santé doivent se conformer à des réglementations strictes en matière de protection des données..[107][114]
- Une connectivité réseau fiable et sécurisée est indispensable pour la transmission des données en temps réel et la surveillance continue des patients.[107]
- Il est indispensable de sensibiliser les patients et les professionnels de la santé aux avantages et à la bonne utilisation des dispositifs IoT pour une mise en œuvre réussie. [107]

2.8 Le triptyque essentiel de la gestion et la fiabilité des données médicales : «Qualité, intégrité et confidentialité»

2.8.1 Mesures pour garantir la qualité et l'intégrité des données :

On peut citer [56] :

- Calibration et maintenance régulières des capteurs et des appareils IoT pour assurer l'exactitude des mesures.
- Utilisation de protocoles de transmission sécurisés pour minimiser les erreurs de transmission et la perte de données.
- Mise en place de mécanismes de contrôle de la qualité des données pour identifier et corriger les anomalies.

2.8.2 Sécurité et confidentialité des données médicales

Pour garantir la sécurité et confidentialité on peut utiliser [57] :

- Cryptage des données pour protéger les informations sensibles pendant la transmission et le stockage.
- Contrôle d'accès strict pour limiter l'accès aux données uniquement aux utilisateurs autorisés.
- Pseudonymisation et anonymisation des données pour protéger l'identité des patients, lorsque possible.

2.9 Revue des travaux connexes

Dans la première partie, nous avons exploré divers aspects de l'Internet des Objets Médicaux (IOMT). Cette partie se concentre maintenant sur l'application de ces connaissances pour améliorer la médecine. Nous essayons d'analyser huit articles qui proposent différentes approches pour intégrer la technologie IoT dans le domaine médical, mettant en lumière les possibilités et les avantages potentiels de l'IoT pour renforcer la surveillance, le diagnostic et la gestion des soins de santé. Ces études ouvrent de nouvelles perspectives pour l'innovation médicale. Voici la présentation de ces articles Dans la suite de ce chapitre :

Bastida et al. [82] ont présenté l'expérience OCARIoT, une solution basée sur l'Internet des objets (IoT) visant à prévenir l'obésité et promouvoir des habitudes de vie saines chez les enfants âgés de 9 à 12 ans. L'architecture proposée par les auteurs intègre plusieurs dispositifs IoT, dont des bracelets de fitness (Fitbit) pour mesurer l'activité physique et le sommeil, ainsi que des balances intelligentes (Fitbit Aria et Beurer) pour surveiller le poids et l'IMC. Le système collecte également des données environnementales des écoles participant aux essais pilotes pour enrichir le modèle de santé. Les informations sont intégrées à la plateforme OCARIoT via une application d'acquisition de données (DAP) et une passerelle API. La plateforme est basée sur une architecture de microservices et inclut un système de support à la décision (DSS) pour évaluer et coacher les comportements de santé des enfants comme la montre la figure 2.1. Cette infrastructure permet une surveillance continue et personnalisée, favorisant l'adoption de comportements sains grâce à une approche éducative.

Le projet a été déployé dans des sites pilotes au Brésil, en Grèce et en Espagne, démontrant une amélioration significative des habitudes alimentaires et de l'activité physique des participants, avec une diminution de la prévalence de l'obésité de 75.5 % dans le groupe d'intervention.

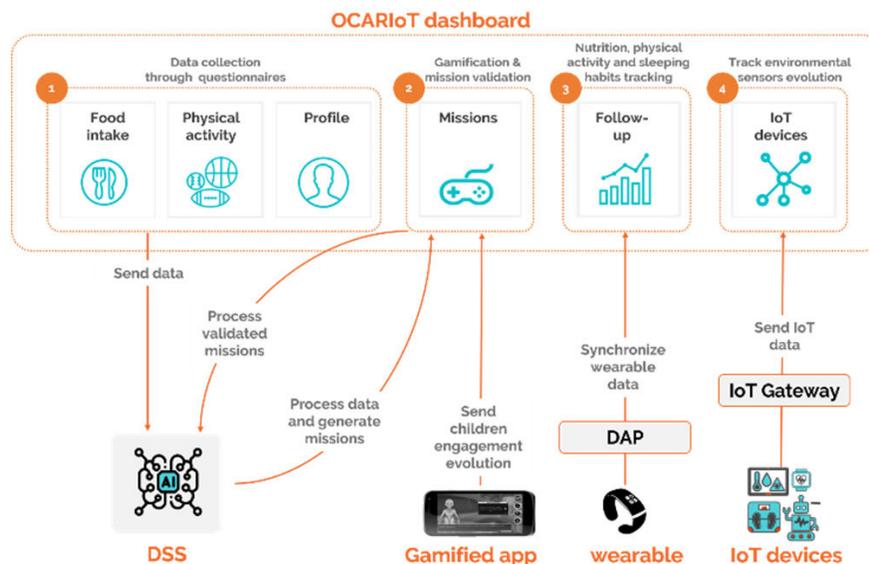


FIGURE 2.1 – Tableau de bord OCARIoT dans l'architecture globale [82].

Sharma et al. [117] motivés par le besoin croissant de solutions de surveillance médicale à distance, proposent une application basée sur l'Internet des Objets (IoT) permet de surveiller en temps réel la santé des patients et de visualiser leur position sur Google Map. Les données de santé, incluant la fréquence cardiaque et la température corporelle, sont collectées par des capteurs et transmises à une base de données Firebase via Wi-Fi ou GPRS. Les auteurs utilisent un Raspberry Pi pour traiter ces données, en exécutant un code en Python sur Raspbian OS. Les données sont accessibles via un portail web sur Microsoft Azure/AWS et une application Android permet aux médecins de suivre les informations sur leur smartphone. Les capteurs analogiques sont intégrés au Raspberry Pi via des modules Arduino ou ADC

pour convertir les données en numérique, tel que présenté dans la figure 2.2. Les auteurs soulignent l'importance de cette solution pour améliorer la surveillance médicale à distance, bien qu'ils identifient des défis en matière de sécurité des données, de gestion de l'énergie et de précision des capteurs. Des améliorations sont suggérées, notamment l'implémentation de protocoles de sécurité, l'optimisation de la consommation d'énergie et l'utilisation de capteurs numériques de haute précision.

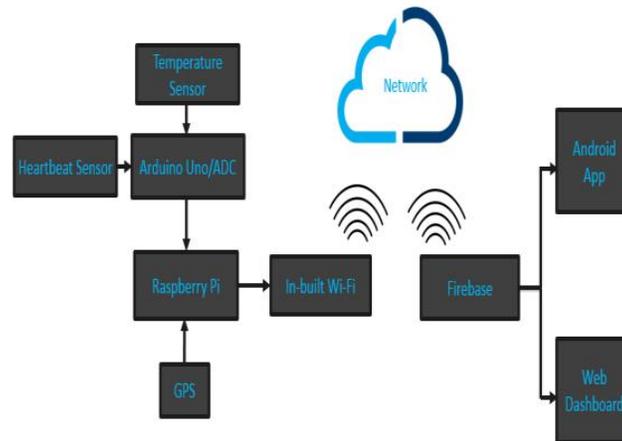


FIGURE 2.2 – Approche de conception [117].

Cet article [124] présente le développement d'une plateforme IoT destinée au diagnostic et au suivi médical à distance des maladies neuromusculaires, en particulier la myopathie de Duchenne.¹, Cette plateforme utilise les capteurs des smartphones, tels que l'accéléromètre et le capteur d'inclinaison, pour surveiller les symptômes les plus courants de cette maladie. La communication entre le patient et le médecin se fait par vidéoconférence, où le médecin pose des questions et demande au patient d'exécuter certaines étapes et gestes afin d'identifier les symptômes de la myopathie de Duchenne. Pour chaque question, un compteur de validité (CV), initialisé à zéro, est incrémenté à chaque réponse positive ("oui"). À la fin de l'évaluation, le médecin fournit un diagnostic indiquant si le patient présente ou non la maladie de Duchenne.

L'article [131], intitulé "Fog-cloud architecture driven Internet of Medical Things framework for healthcare monitoring", propose une approche basée sur l'utilisation de la technologie de l'Internet des objets pour la prédiction du diabète, en utilisant le Fog computing et le cloud computing, ainsi que des techniques d'apprentissage automatique pour la prise de décision. Cette approche comprend un ensemble de huit capteurs corporels implantés dans le corps pour mesurer diverses données telles que le glucose, la pression artérielle, l'indice de masse corporelle, le score génétique de diabète, le nombre de grossesses, l'épaisseur de la peau et d'autres paramètres. Ces huit capteurs sont interconnectés via le protocole Ad

1. La myopathie de Duchenne est une maladie génétique dégénérative caractérisée par une faiblesse musculaire progressive.

hoc On-Demand Distance Vector (AODV) selon la norme de communication IEEE 802.15.6, avec une méthode d'accès multiple Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Les données collectées par ces capteurs sont centralisées par un hub coordonnateur, formant un réseau WBAN (Wireless Body Area Network). Le hub transmet ensuite ces données au composant fog le plus proche, situé à proximité du WBAN, en utilisant la logique floue², une approche approximative et non probabiliste. Les données sont également envoyées à une passerelle, en tenant compte de la priorité des données des capteurs implantés selon la configuration définie. La passerelle transmet ces données vers un cloud situé plus loin du WBAN, avec lequel elle est la seule à pouvoir communiquer en raison de la portée de communication limitée du cloud. Les données stockées dans le fog sont traitées localement et sont généralement de nature en temps réel, tandis que les données stockées dans le cloud sont destinées à des analyses plus approfondies et à un stockage à long terme.

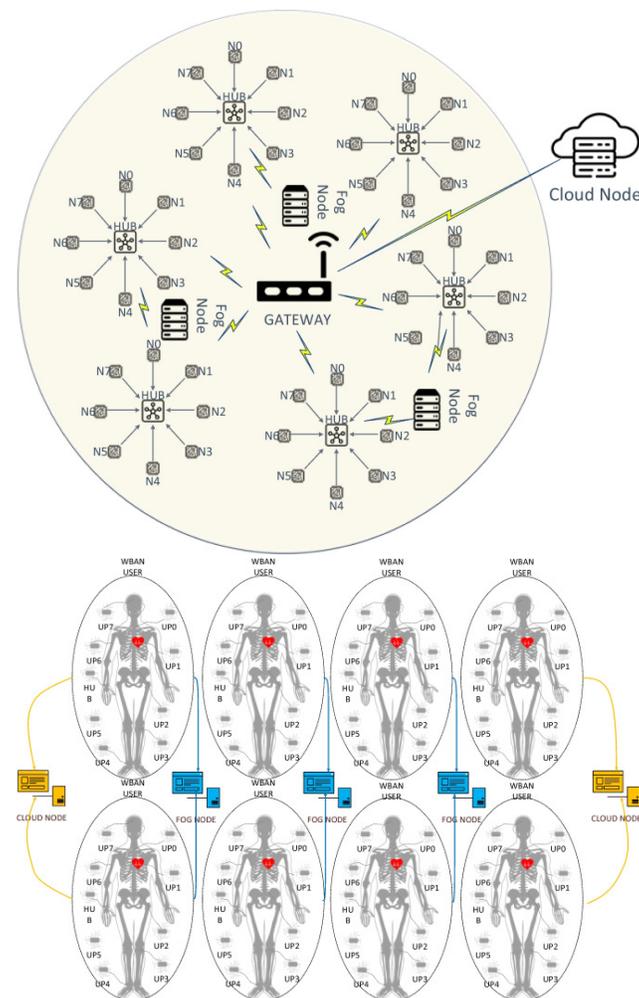


FIGURE 2.3 – La topologie et l'architecture du framework IoMT proposé[131].

2. Logique floue est une approche approximative et non probabiliste fonctionne selon les étapes suivantes : d'abord, la fuzzification qui calcule un degré d'appartenance en comparant par exemple les valeurs de glycémie déjà stockées avec celles capturées. Ensuite, la base de connaissances applique les règles (si alors). Après vient l'étape du moteur d'inférence utilisant la logique (ET – OU). Enfin, le défuzzificateur transforme le résultat obtenu.

L'article [105] intitulé "Conception et implémentation d'une application IoMT pour le diagnostic et le suivi du diabète" propose une approche pour le suivi des patients diabétiques, avec la création de l'application "Diabloop". Cette application permet de mesurer le taux de glycémie dans le sang et de surveiller l'évolution du diabète. Elle intègre un algorithme présenté dans la figure 2.5 capable de prendre des décisions de manière autonome pour assister le patient et envoyer des notifications au médecin ou à un proche en cas de besoin.

L'architecture de communication illustrée dans la figure 2.4 permet à un capteur de glycémie d'envoyer ses données via Bluetooth à l'application Diabloop installée sur un smartphone. Ces données sont ensuite transmises à une passerelle, puis vers le cloud, où le médecin peut consulter les informations des patients dès qu'une connexion Internet est disponible. Le résultat final est alors transmis à l'utilisateur.

Diabloop adopte une méthode de synchronisation asynchrone pour mettre à jour le serveur dès qu'une connexion est établie, permettant ainsi au serveur de recevoir les données les plus récentes des capteurs

Les créateurs de Diabloop ont modélisé l'application en utilisant le langage UML et l'ont implémentée en PHP. Le capteur utilisé est un dispositif mesurant la concentration de glucose dans le sang.

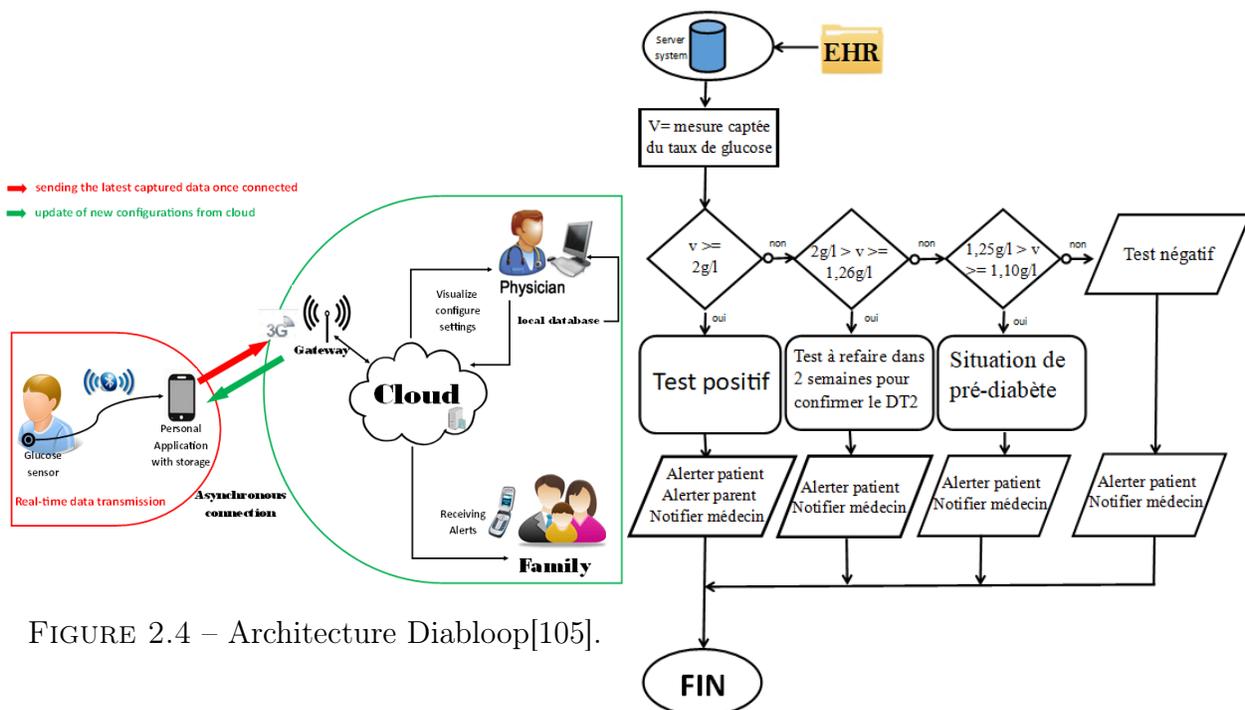


FIGURE 2.4 – Architecture Diabloop[105].

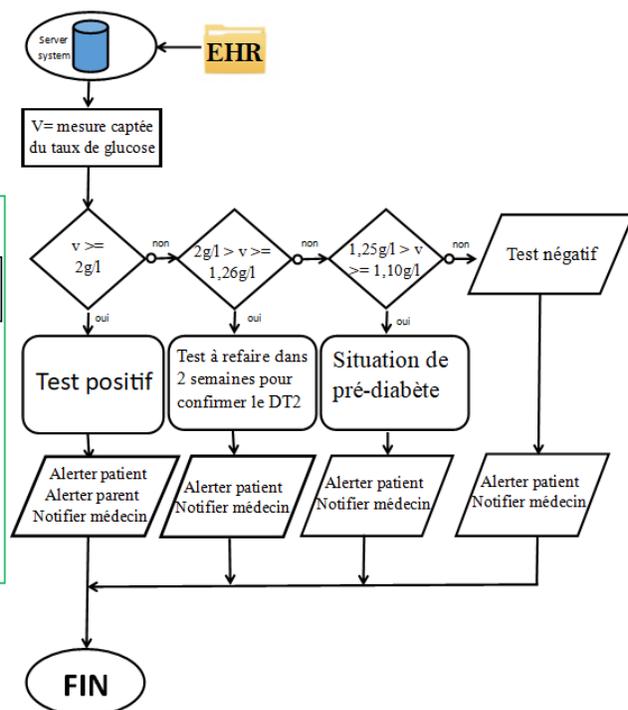


FIGURE 2.5 – Algorithme de base de Diabloop[105].

EcoHealth [104] est une plateforme web middleware conçue pour l'intégration transparente de divers dispositifs de santé électronique (e-health). Elle vise à améliorer la surveillance continue des patients en temps réel en connectant différents capteurs corporels (fréquence cardiaque, pression artérielle) à une interface web conviviale. Les données collectées sont

traitées, stockées de manière sécurisée et présentées de manière intuitive aux médecins pour une prise de décision éclairée. L'architecture modulaire et flexible d'EcoHealth facilite l'ajout de nouvelles fonctionnalités et l'intégration de futurs dispositifs médicaux, assurant ainsi son adaptabilité aux évolutions technologiques et aux besoins changeants des utilisateurs.

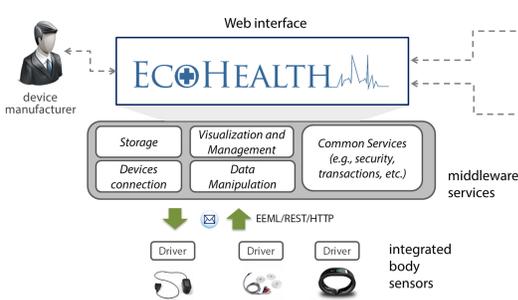


FIGURE 2.6 – L'architecture EcoHealth[104].

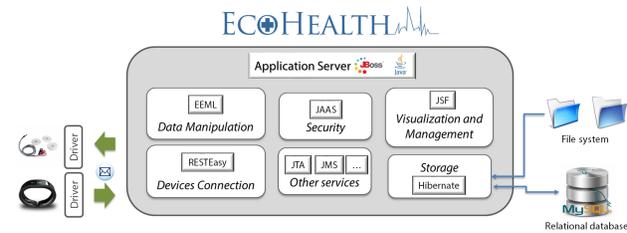


FIGURE 2.7 – Les technologies utilisées pour implémenter les modules EcoHealth[104]

Dans l'approche proposée par l'article [130] intitulé "A Novel Approach for Multi-channel Epileptic Seizure Classification Based on Internet of Things Framework Using Critical Spectral Verge Feature Derived from Flower Pollination Algorithm", un capteur sans fil de type Wireless Sensor Network (WSN) est utilisé pour enregistrer les signaux électro-encéphalographiques (EEG) du patient. Placé sur le cuir chevelu du patient, ce capteur capture les signaux EEG, qui sont ensuite transmis via Bluetooth ou Wi-Fi à une application Android installée sur le smartphone du patient. Cette application envoie les données EEG au Cloud pour traitement en utilisant des protocoles sécurisés comme HTTP/HTTPS. Le traitement des données EEG est réalisé sur deux serveurs Cloud : un Raspberry Pi 4, utilisé pour certaines tâches de traitement et de communication, et Amazon EC2, un service de Cloud computing pour des applications à grande échelle.

Les étapes de traitement incluent d'abord la segmentation des signaux EEG en segments de 1040 points de données (4 secondes) sans chevauchement, suivie d'un prétraitement basé sur la méthode de filtrage adaptatif basé sur multirésolution (MRAF) pour réduire les artefacts physiologiques. Ensuite, un point de verge spectrale critique (CSV) est extrait en calculant la densité spectrale de puissance (PSD) moyenne pour chaque bande de fréquence et en comparant avec le point de fréquence maximal (PSD segment).

Si la PSD segment est supérieure à la PSD moyenne, il est identifié comme un point de verge spectrale, puis optimisé via l'algorithme de pollinisation des fleurs (FPA) pour obtenir un point de verge spectrale critique (CSV), représentant les oscillations de haute fréquence. Une réduction de dimensionnalité est effectuée avec l'analyse en composantes principales (PCA), suivie par la classification des segments EEG en "crise" ou "non crise" à l'aide de méthodes comme k-NN, SVM ou K-Means.

Les résultats du traitement sont ensuite envoyés au smartphone du patient et du médecin en utilisant le protocole REST HTTP pour la transmission des données et le protocole Message Queuing Telemetry Transport (MQTT) pour les notifications en temps réel sur les changements anormaux détectés dans les signaux EEG. La gestion des accès et la sécurité sont

assurées par Active Directory (AD), utilisé pour installer et gérer l'application de traitement sur les serveurs Cloud.

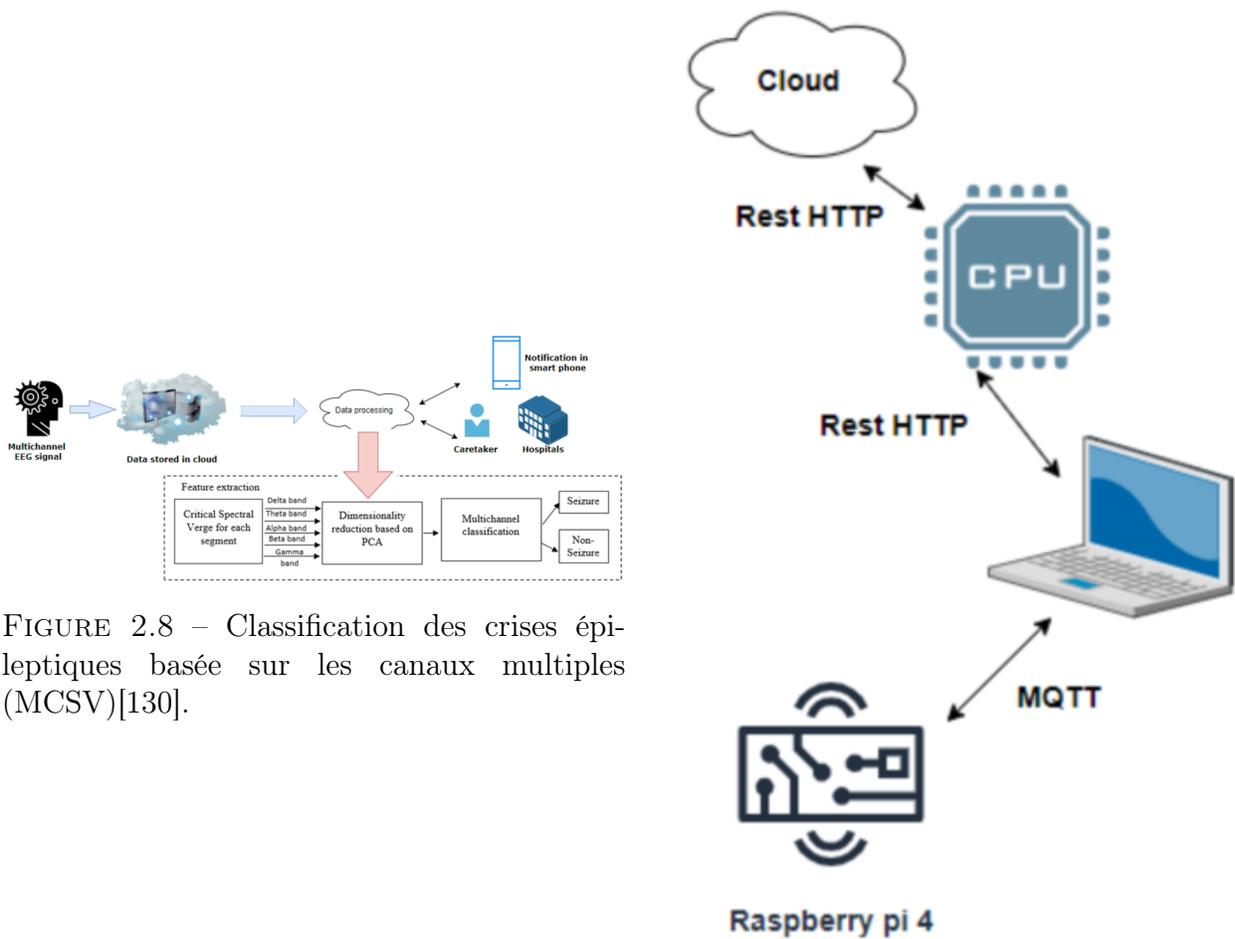


FIGURE 2.8 – Classification des crises épileptiques basée sur les canaux multiples (MCSV)[130].

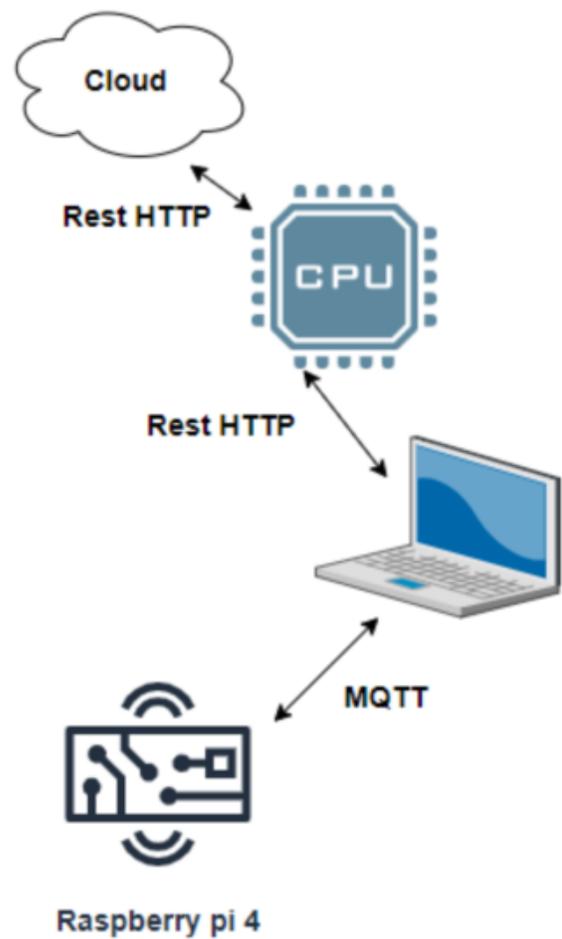


FIGURE 2.9 – Architecture système de la classification proposée des crises épileptiques[130].

L'article [87] intitulé "Using IoT technologies to develop a low-cost smart medicine box" présente une approche qui propose une boîte à médicaments intelligente à faible coût, conçue pour rappeler aux utilisateurs de prendre leurs médicaments à temps. son architecture IoT basée sur le fog, organisée en trois couches distinctes illustrées dans la figure2.10. La couche comprend des appareils intelligents dotés de capacités d'identification, de détection et d'action Elle est équipée de plusieurs composants, notamment un microcontrôleur ESP8266, Une boîte à usage multiple, quatre LEDs de couleurs différentes pour indiquer visuellement les moments de la journée où les médicaments doivent être pris, un buzzer pour les alertes sonores, et un capteur d'inclinaison pour détecter les mouvements de la boîte.

La couche edge/fog, est constituée de passerelles Raspberry Pi modèle 3, équipé du logiciel Eclipse Kura. Connecter par MQTT. Enfin, la troisième couche est la couche Cloud, basé sur Google Virtual Machine et utilisant la plateforme Cloud IoT Eclipse Kapua.

Le processus commence par la configuration initiale des horaires de prise des médica-

ments via une interface web, qui envoie ces configurations au Cloud. Le Cloud transmet ensuite ces configurations à la passerelle via MQTT. La passerelle envoie les horaires configurés à la boîte à médicaments, toujours via MQTT. Lorsqu'un patient ouvre la boîte, le capteur d'inclinaison détecte cette action et envoie un signal à la passerelle, qui transmet cette information au Cloud pour mettre à jour les journaux de prise de médicaments. Le patient et les soignants peuvent ensuite visualiser l'état des médicaments et les prises effectuées via une interface web^{2.11} connectée au Cloud.

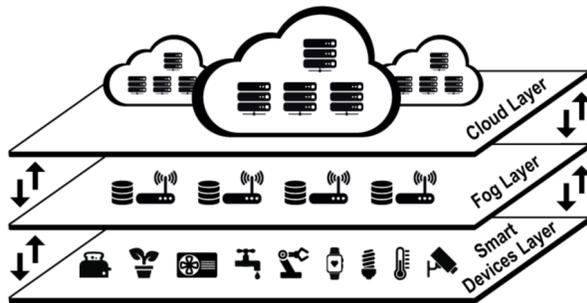


FIGURE 2.10 – Architecture générale multi-couche[87].

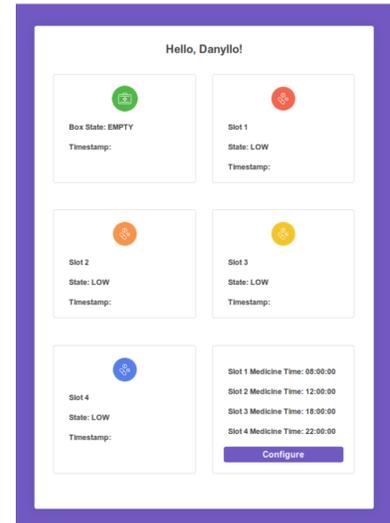


FIGURE 2.11 – L'interface utilisateur[87].

TABLEAU 2.2 – Comparaison des articles IoT pour la santé.

Article Critère	IoT pour la Surveillance de la Santé	OCARIoT pour la Prévention de l'Obésité	DiabLoop pour le Diagnostic et le Suivi du Diabète	Plateforme IoT pour l'aide au Diagnostic des Maladies Neuromusculaires	Cadre IoMT basé sur l'architecture Fog-Cloud pour la surveillance de la santé	Plateforme Web pour l'interconnexion de capteurs corporels et l'amélioration des soins de santé	Boîte à médicaments intelligente à faible coût utilisant l'IoT	Classification IoT des crises d'épilepsie basée sur MCSV
Objectif	Surveillance en temps réel de la santé des patients	Prévention de l'obésité chez les enfants	Diagnostic et suivi du diabète	Aide au diagnostic des maladies neuromusculaires	Surveillance de la santé et prédiction du diabète	Intégration de divers capteurs corporels pour la surveillance continue des patients	Rappel de prise de médicaments au bon moment	Classification automatique des crises d'épilepsie et localisation de l'activité épileptique dans la région focale du cerveau
Population ciblée	Patients en général	Enfants âgés de 9 à 12 ans	Patients diabétiques ou à risque	Patients atteints de maladies neuromusculaires	Patients diabétiques	Patients nécessitant une surveillance continue	Utilisateurs de médicaments	Patients épileptiques

Capteurs utilisés	Pulse sensor, capteur de température	Bracelets Fitbit, balances intelligentes (Fitbit Aria et Beurer), capteurs environnementaux	Accéléromètre, gyroscope, podomètre, caméras (smart-phone)	Capteurs ECG, capteurs corporels sans fil (WBAN)	Capteur de pression artérielle, autres capteurs biométriques	Non spécifié	Non spécifié	Technologie de capteurs sans fil (WSN) pour enregistrer les signaux EEG
Technologies utilisées	Wi-Fi, GPRS, HTTP	Bluetooth Low Energy, REST API	Non spécifié	Non spécifié (smart-phone comme outil de communication)	IEEE 802.15.6, AODV	HTTP	Wi-Fi	Réseau de capteurs sans fil (WSN)
Techniques de sécurité	Connexions SSL, certificats	SSL	Connexions SSL, OAuth 2.0	Non spécifié	Non spécifié	Non spécifié	Non spécifié	Non spécifié
Base de données	Firebase	Non spécifié (plateforme de microservices)	MySQL Cloud	Non spécifié	MySQL	Non spécifié	Non spécifié	Non spécifié

Plateforme	Microsoft Azure/AWS, application Android	Architecture de microservices, DAP, passerelle API, application gamifiée, tableau de bord	PHP, application web, application mobile (en développement)	Smartphone, Cloud	Fog, Cloud	Java, JBoss, RESTEasy, JAX-RS	ESP8266, Raspberry Pi, Eclipse Kura, Google VM, Eclipse Kapua	Smartphone Android, Raspberry Pi 4, Amazon Elastic Compute Cloud (EC2)
DSS (Decision Support System)	Non mentionné	Fournit des évaluations comportementales et des plans de coaching personnalisés	Système intelligent avec base de connaissances pour aider à la prise de décision	Protocoles de diagnostic (en développement)	Logique floue (brouillard), apprentissage automatique (cloud)	Non mentionné	Non mentionné	Un système de surveillance des crises d'épilepsie basé sur le cloud pour aider les neurologues
DAP (Data Acquisition App)	Non mentionné	Application pour intégrer les données des dispositifs IoT	Application mobile pour la collecte de données (en développement)	Application mobile (en développement)	Non mentionné	Non mentionné	Interface web (NodeJS, Heroku)	Application pour smartphone Android pour la capture et la transmission de données EEG

Méthodologie	Surveillance continue via capteurs et communication réseau	Collecte de données, évaluation comportementale, et coaching personnalisé	Mesure régulière de la glycémie, envoi de notifications, prescriptions médicamenteuses	Collecte de données via capteurs smartphone, comparaison avec des références	Analyse des données WBAN à l'aide de la logique floue (brouillard) et de l'apprentissage automatique (cloud)	Collecte de données via capteurs, transmission via HTTP, structuration avec EEML	Configuration des horaires de prise de médicaments via une interface web, rappels visuels (LED) et sonores (buzzer) à l'heure prévue	Prétraitement du signal EEG, extraction de caractéristiques optimisée par l'algorithme de pollinisation des fleurs (FPA), classification multicanal à l'aide de machines à vecteurs de support (SVM)
---------------------	--	---	--	--	--	--	--	--

Résultats observés	Surveillance en temps réel de la santé des patients	Amélioration des habitudes alimentaires et physiques des enfants	Facilitation de la gestion du diabète, suivi à distance	Outil de pré-diagnostic, suivi à distance	Prédiction du diabète (précision de 64 % dans le brouillard, 89,5 % dans le cloud)	Visualisation des données (ex : fréquence cardiaque, pression artérielle) via une interface web, alertes en cas d'anomalies	Prototype fonctionnel, démonstration de faisabilité	Amélioration de la classification des crises d'épilepsie, localisation précise de l'origine des crises
Déploiement	Non spécifié	Déployé dans des écoles au Brésil, en Grèce et en Espagne	Non spécifié	Non spécifié	Non spécifié	Non spécifié	Non spécifié	Non spécifié
Niveau	Non spécifié	Communauté	Clinique	Clinique	Clinique et à distance	Non spécifié	Non spécifié	Clinique

2.10 Analyse et Discussion

Les travaux étudiés montrent l'efficacité de l'IoT dans le domaine de la santé, chaque projet porte son attention sur des défis spécifiques avec des solutions créatives pour les patients et les professionnels de santé. Alors à travers l'analyse de ces travaux de recherche, nous avons pu illustrer quelques applications médicales variées de l'IoT. Par exemple, certains projets se concentrent sur la surveillance en temps réel de la santé des patients, permettant une intervention rapide en cas de détérioration de leur état. D'autres visent à prévenir des maladies comme l'obésité infantile en combinant des capteurs de suivi de l'activité physique avec des plateformes d'analyse comportementale. Des systèmes dédiés au diagnostic et au suivi de maladies chroniques comme le diabète ou les maladies neuromusculaires montrent comment l'IoT peut faciliter la gestion quotidienne de ces conditions, tout en offrant un soutien décisionnel aux professionnels de santé.

Ces travaux nous ont permis de mieux comprendre les diverses approches et technologies employées pour résoudre des problèmes critiques en santé, et ils ont également servi de référence pour créer une architecture IoT de la gestion du stock de sang qui s'inspire de ces approches, en intégrant des technologies avancées et des méthodologies éprouvées pour la mise en place d'une plateforme IoT robuste pour une gestion des stocks de sang sécurisée, performante et adaptable.

2.11 Conclusion

Ce chapitre a exploré l'impact de l'Internet des Objets (IoT) dans le secteur médical, mettant en évidence comment cette technologie révolutionne la gestion des soins de santé. En intégrant divers capteurs, réseaux et plateformes, l'IoT permet une surveillance continue, une gestion efficace des maladies chroniques et une amélioration de la qualité des soins prodigués aux patients.

Les avancées technologiques et les principes fondamentaux de l'IoT serviront de base à la conception de notre propre architecture pour la gestion du stockage de sang, qui sera détaillée et explorée en profondeur dans le chapitre suivant.

Plateforme IoT pour le stockage et la gestion automatisée du de sang

3.1 Introduction

Dans le domaine médical contemporain, la gestion efficace des banques de sang constitue un pilier essentiel pour assurer des soins de santé de haute qualité. Toutefois, les méthodes traditionnelles de gestion manuelle présentent des défis considérables, notamment en termes de précision, de fiabilité et de réactivité. Ces défis peuvent potentiellement entraver la disponibilité opportune et adéquate du sang nécessaire pour répondre aux besoins urgents des patients, qu'il s'agisse d'interventions chirurgicales critiques, de traitements d'urgence ou de soins intensifs.

Pour répondre à ces défis, une architecture avancée basée sur l'Internet des Objets (IoT) est spécifiquement conçue pour révolutionner la gestion des banques de sang. Cette architecture repose sur l'intégration d'un ensemble de capteurs intelligents, de plateforme cloud et de systèmes de communication sécurisés. Grâce à l'utilisation de ces technologies avancées, il est possible de développer des systèmes du stockage automatisés et en temps réel pour le sang, capables de surveiller en permanence des paramètres environnementaux essentiels tels que la température, l'humidité et les niveaux de stock.

Dans ce qui suit, une étude est entreprise pour examiner cette architecture et mieux évaluer ses effets pratiques.

3.2 Système automatisé de gestion du sang : une nécessité pour des soins de santé modernes

Dans de nombreux hôpitaux ,L'absence de systèmes automatisés de gestion du stockage de sang constitue un obstacle majeur à la garantie d'un approvisionnement en sang sûr et de qualité en temps réel. Les approches manuelles actuelles, souvent inefficaces et sujettes aux erreurs humaines, entravent l'accès immédiat aux produits sanguins vitaux, particulièrement en cas d'urgence.

Face à ce défi, l'implémentation d'une chambre automatisée de gestion et du stockage de sang s'impose comme une solution indispensable pour répondre aux besoins urgents des

patients et garantir un accès immédiat à tous les groupes sanguins. En s'appuyant sur la technologie IoT et les technologies de pointe de surveillance et de communication, cette solution innovante optimise la gestion des stocks de sang, renforce la sécurité des services médicaux, et contribue ainsi à l'accroissement de la fiabilité des systèmes de santé modernes. [16]

Dans la partie suivante, on va présenter une étude de l'architecture IoT est proposée ,dans le but d'optimiser et automatiser la gestion des stocks de sang, en mettant en avant sa valeur ajoutée et ses composants, et en soulignant son importance et ses applications.

3.3 Aperçu global de l'architecture proposée

Dans le cadre de l'amélioration continue de la gestion des stocks de sang, nous proposons une infrastructure automatisée du stockage répartie sur quatre niveaux. Cette infrastructure modulaire se compose de quatre étages dédiés, chacun étant conçu pour stocker un type de sang spécifique en respectant la distinction Rh+ et Rh-. Par exemple,le niveau 1 peut accueillir les poches de sang A+ et A- dans chaque case de niveau 1, tandis que le niveau 2 est dédié aux poches B+ et B-. Chaque case est équipée d'un capteur de pression Honeywell 26PC assurant une surveillance précise du volume de sang restant, permettant ainsi un contrôle optimal des stocks.

La chambre est également dotée de capteurs d'humidité et de température SI7021, ainsi que d'un capteur de tension électrique ZMPT101B connecté à l'alimentation principale pour les cas de coupures de courant. Tous ces capteurs sont reliés à un microcontrôleur ESP32, qui lit et traite les données. Le microcontrôleur est connecté à un Fog Computing via un réseau de communication LoRa. Fog Computing transmet ensuite les données à une passerelle LoRaWAN, qui envoie les informations vers le cloud.

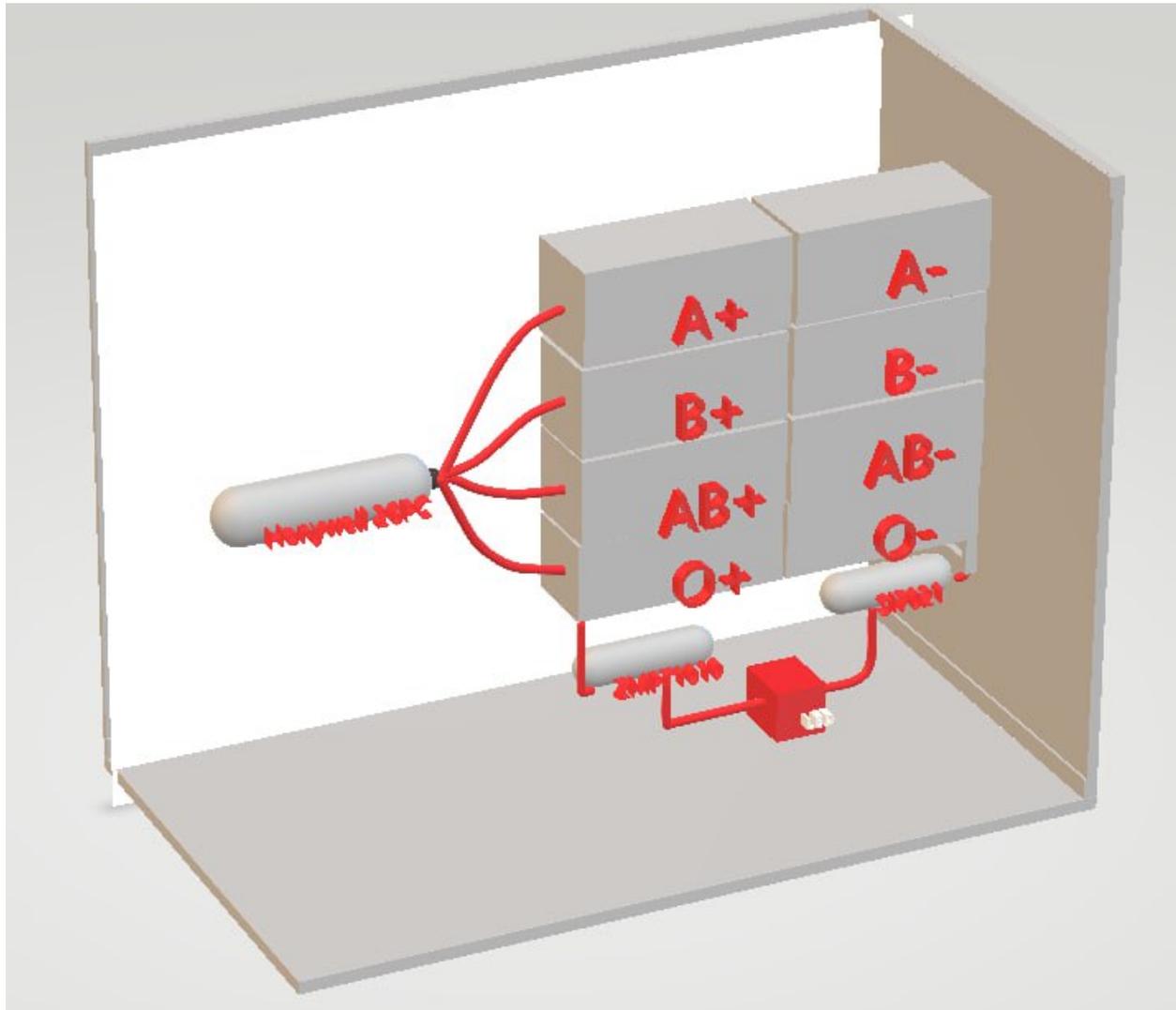


FIGURE 3.1 – Chambre automatisée de stockage de sang en 3D.

Le cloud héberge une interface utilisateur permettant d'envoyer des notifications par SMS via Firebase (Google Cloud Firebase) ou Twilio en cas de demande urgente de sang. L'objectif principal de cette approche est de notifier les donateurs de sang lorsque le capteur de pression détecte que le volume d'un type de sang est de 40% à 60%. Un appel d'urgence est déclenché si le seuil atteint 40% [17].

à l'extérieur de la chambre Une diode électroluminescente (LED) rouge s'allume pour signaler toute condition critique, telle que des anomalies de température, d'humidité ou de volume de sang. Cette LED sert également d'alerte visuelle en cas de coupure de courant, avertissant immédiatement le responsable médical de la situation. Connectée à un microcontrôleur, la LED assure une surveillance constante des paramètres critiques, permettant une intervention rapide en cas de problème [18][19].

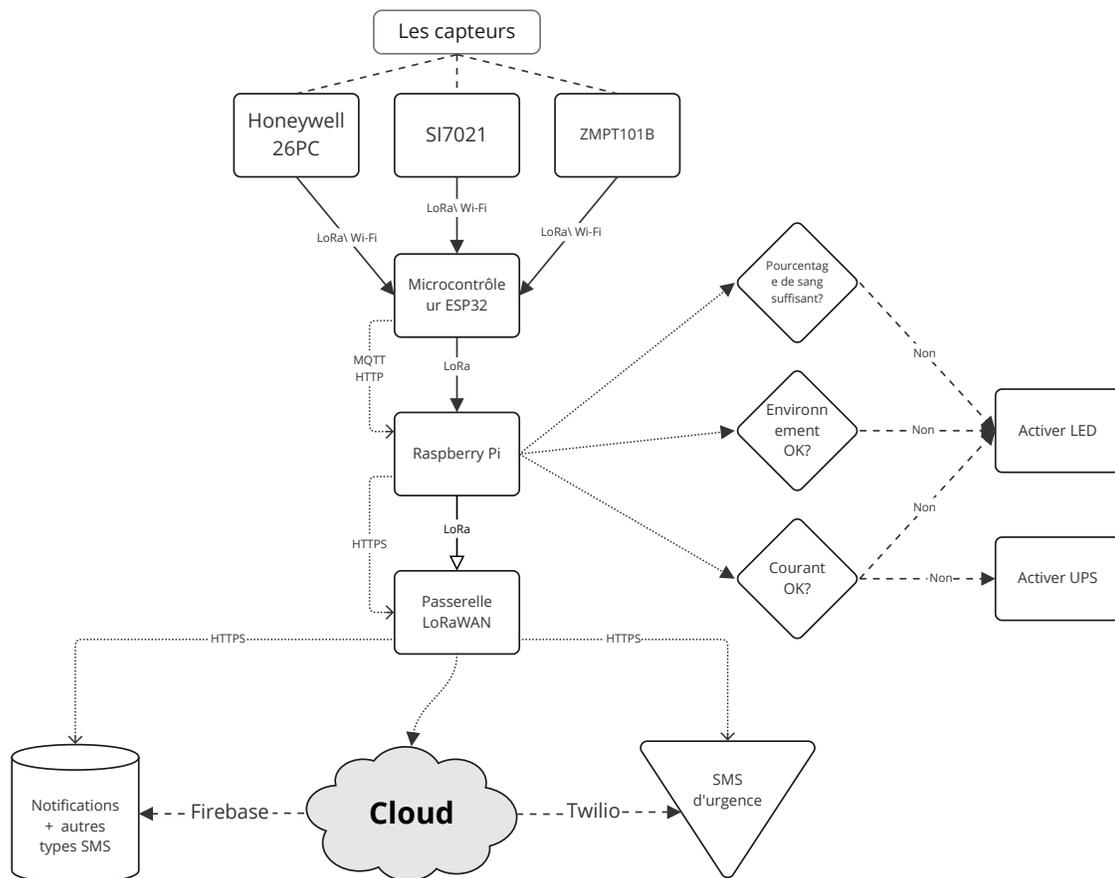


FIGURE 3.2 – Diagramme représentatif de l'architecture pour la gestion automatisée du stockage du sang.

3.4 Justification de choix de chaque composant intégré

3.4.1 Capteur de pression (Honeywell 26PC)

Ce capteur 3.3 [127][129] [83] est utilisé pour mesurer la pression exercée par le poids du sang sur une surface, permettant ainsi le calcul du volume de sang restant. Ses caractéristiques sont les suivantes :

- ▷ Il fonctionne dans une large gamme de températures, allant de -40°C à $+85^{\circ}\text{C}$.
- ▷ Facile à intégrer directement avec un microcontrôleur via une interface analogique.
- ▷ Il fournit des quantités de sang presque exactes grâce à sa capacité de mesure de pression.
- ▷ Sa faible consommation d'énergie est particulièrement avantageuse pour une surveillance continue avec une batterie de 1000 mAh à 3.3 v pourrait théoriquement fonctionner pendant environ de 200 heures, soit un peu plus de 8 jours.
- ▷ Il est suffisamment sensible pour détecter de petites variations de volume.



FIGURE 3.3 – Capteur de pression (Honeywell 26PC)[34].

Contrairement aux capteurs de poids : qui peuvent être affectés par des vibrations ou des mouvements externes (par exemple l'ouverture de la porte), ce capteur de poids est utilisé pour des charges importantes. De plus, l'installation de capteur de pression est plus simple et il peut être plus économique.

3.4.2 Capteur d'humidité et de la température (SI7021)

SI7021 [103] est un composant électronique couramment utilisé pour mesurer à la fois l'humidité relative et la température ambiante dans diverses applications, a une faible consommation d'énergie typiquement 150 milli ampère, ce qui le rend idéal pour une utilisation dans des applications alimentées par batterie ou soucieuses de l'efficacité énergétique.

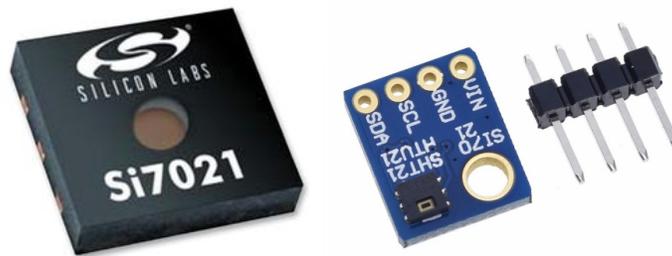


FIGURE 3.4 – Capteur SI7021)[35].

3.4.3 Capteur de tension électrique (ZMPT101B)

ZMPT101B [78] est un module conçu pour mesurer la tension alternative (AC) avec une grande précision généralement de 0 à 250V AC. Il est couramment utilisé dans des applications de surveillance et de contrôle de l'alimentation électrique. Il dispose de sorties analogiques qui peuvent être facilement lues par un ADC (convertisseur analogique-numérique) sur des microcontrôleurs comme l'ESP32. Le module est conçu pour consommer très peu d'énergie, ce qui est bénéfique pour les applications où l'efficacité énergétique est importante. Le ZMPT101B Offre une isolation galvanique entre le circuit de mesure et le circuit de puissance, assurant ainsi la sécurité et la protection des composants électroniques sensibles, y compris les autres capteurs mentionnés précédemment.

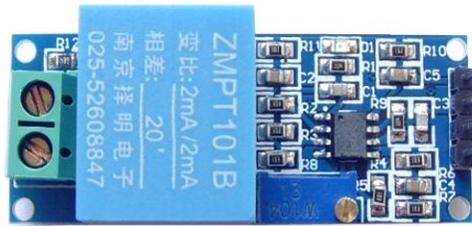


FIGURE 3.5 – Capteur de tension électrique (ZMPT101B) [78].

3.4.4 Microcontrôleur ESP32

ESP32 [96] [20] [21] est un ordinateur miniature intégrant un processeur, une mémoire et des périphériques d'entrée/sortie programmables, il est capable de lire les données des capteurs intégrés. Programmable à l'aide de plateformes telles qu'Arduino, l'ESP32 est largement plébiscité dans le domaine des projets IoT pour ses performances élevées et sa faible consommation d'énergie. Il constitue ainsi un élément important dans de nombreuses applications de gestion et de contrôle intelligentes, telles que la gestion automatisée du stockage de sang.

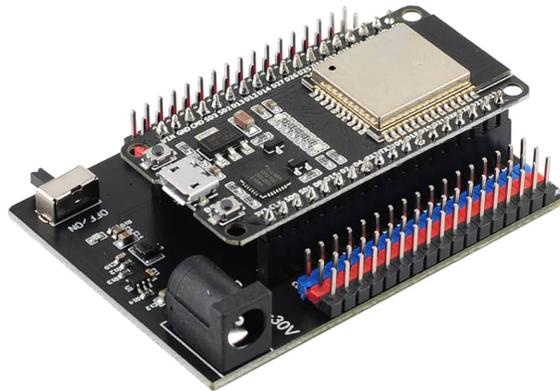


FIGURE 3.6 – Microcontrôleur ESP32 [34].

3.4.5 Fog Computing (Raspberry Pi)

Le concept de "Fog Computing" [22] [90] décrit une architecture informatique décentralisée où le traitement des données se déroule à proximité de leur source, plutôt que de les transférer vers un cloud centralisé pour être traitées. Dans le cadre de notre projet IoT, nous avons choisi le Raspberry Pi comme une solution privilégiée en raison de sa polyvalence, de sa fiabilité et de sa disponibilité.



FIGURE 3.7 – Raspberry Pi[23].

Voici quelques-unes des raisons pour lesquelles le Raspberry Pi est une option et choix idéale pour notre architecteur :

- ▷ Il présente un coût abordable (solution économique).
- ▷ Une vaste communauté de soutien et de développement, le Raspberry Pi offre un accès aisé à des ressources et à une assistance en cas de besoin.
- ▷ Raspberry Pi offre une exécution locale des tâches nécessaires avant d'envoyer les données vers le cloud.
- ▷ Facilité de Configuration.

3.4.6 Passrelle (LoRAWAN)

Une passerelle LoRaWAN est un dispositif qui permet la communication entre les dispositifs IoT équipés de la technologie LoRa et un réseau LoRaWAN. Elle agit comme un pont entre les dispositifs IoT et le réseau, facilitant ainsi la transmission des données sur de longues distances avec une faible consommation d'énergie.[25][24][102]

3.4.7 Cloud

Cloud [27] [28] dans cette architecture permet de [26] :

- Centraliser les données collectées par les capteurs en un seul endroit accessible. Cela facilite la gestion et l'analyse des informations en temps réel.
- Sécuriser et rendre accessibles les données critiques du stockage de sang, tout en automatisant les notifications et en offrant une flexibilité et une évolutivité accrues.
- Les données stockées dans le cloud peuvent être consultées de n'importe où et à tout moment.
- Automatisation des Notifications par l'utilisation de Firebase et Twilio. Par exemple, si le volume de sang tombe en dessous d'un certain seuil, des notifications peuvent être envoyées aux donneurs de sang via SMS ou appels téléphoniques.

- Le Cloud offre la possibilité d'ajuster facilement les ressources en fonction des besoins. Si la quantité de données à traiter augmente, le système peut évoluer sans nécessiter de modifications matérielles importantes.
- Utiliser le Cloud peut réduire les coûts associés à la gestion et à la maintenance de l'infrastructure physique. Les frais sont souvent basés sur l'utilisation, permettant de payer uniquement pour les ressources consommées.
- Garantie la Sécurité des Données.

3.4.8 Twilio

Twilio [29] est une plateforme de communication Cloud qui permet aux développeurs de créer des applications de communication intégrant la messagerie, les appels vocaux, la vidéo et d'autres services de communication. Twilio fournit une Application Programming Interface (API) facile à utiliser qui permet d'intégrer ces fonctionnalités directement dans les applications web et mobiles. Voici quelques caractéristiques de Twilio :

- Messagerie SMS et Multimedia Messaging Service (MMS) : Envoi et réception de messages texte.
- Réalisation et réception d'appels téléphoniques via le Cloud.
- Obtention de numéros de téléphone pour divers pays.
- Application Programming Interface (API) et Software Development Kit (SDK) pour ajouter des capacités de communication aux applications.

Dans notre architecture on utilise Twilio pour les appels d'urgence. Il permet d'envoyer des notifications immédiates et critiques aux donneurs de sang ou au personnel médical via des appels téléphoniques ou des SMS, assurant une communication rapide et sûr en cas de besoin urgent.

3.4.9 Firebase

Firebase [30] est une plateforme de développement d'applications mobiles et web fournie par Google. Elle propose une variété de services et d'outils pour aider les développeurs à créer des applications de haute qualité, à croître leur base d'utilisateurs et à gagner plus de revenus. Voici quelques-unes des principales fonctionnalités de Firebase :

- Synchronisation des données entre les clients en temps réel.
- Système d'authentification sécurisé pour les utilisateurs.
- Envoi de notifications push aux utilisateurs.
- Suivi et analyse du comportement des utilisateurs.

Firestore est utilisée pour les notifications régulières. Elle gère les alertes envoyées aux donneurs de sang lorsque le volume de sang atteint un seuil prédéfini, en utilisant des services tels que les notifications Push et les bases de données en temps réel pour assurer une communication efficace et fiable. Firestore est choisie pour les notifications non urgentes parce qu'elle est gratuite, mais il dépend de la connexion Internet, ce qui peut parfois entraîner des cas où les notifications SMS ne sont pas reçues.

Remarque : dans l'architecture proposée, il est possible d'utiliser uniquement Firestore pour la gestion des notifications, qu'elles soient normales ou urgentes, supprimant ainsi la nécessité d'intégrer Twilio.

3.5 Fonctionnement de l'architecture

3.5.1 Collecte des données

Les données captées par les capteurs de pression Honeywell 26PC, Capteur d'humidité et de température SI7021 et Capteur de tension électrique ZMPT101B sont envoyées au microcontrôleur ESP32 :

3.5.1.1 Capteur de pression

Honeywell 26PC est connecté au ESP32 via le module émetteur-récepteur radio le RFM95 utilisant le réseau de connexion longue range (LoRa) ou wifi, via l'interface Serial Peripheral Interface (SPI).

Principe de fonctionnement de Honeywell 26PC [15]

1. Détection de la pression[128] :
 - (i) Honeywell 26PC comme tous les capteurs de pression possèdent une partie appelée diaphragme, qui est flexible.
 - (ii) Lorsque le sang est présent dans le réservoir, il exerce une force sur ce diaphragme.
 - (iii) La force (ou pression) exercée par le sang fait que le diaphragme se déforme légèrement.
2. Conversion de la déformation en signal électrique :
 - (i) Des éléments appelés jauges de contrainte sont attachés au diaphragme.
 - (ii) Lorsque le diaphragme se déforme, ces jauges changent de résistance électrique.
 - (iii) Ce changement de résistance est converti en un signal électrique, tel qu'une tension.
3. Lecture de signal[31] :
 - (i) Le signal électrique est amplifié et envoyé à un microcontrôleur comme l'ESP32.
 - (ii) Le microcontrôleur lit ce signal électrique et le convertit en une valeur de pression.
4. Conversion de la pression en quantité de sang : Pour utiliser la pression mesurée pour déterminer la quantité de sang, nous devons effectuer un étalonnage. Voici comment cela fonctionne :

(a) **Etalonnage**

Etalonnage est un processus expérimental essentiel pour garantir la précision des mesures prises par les capteurs. Il permet de convertir des valeurs brutes (comme la pression mesurée) en unités utiles (comme la quantité de sang). Ce processus se fait généralement une fois de manière expérimentale, puis la relation établie est codée dans le microcontrôleur pour une utilisation continue.

- i. **Mesure avec Quantités Connues** : On remplit le réservoir de sang avec des quantités connues et on mesure la pression correspondante dans cette architecture on propose de faire par exemple 80 litres dans chaque case de chaque type de sang et chaque sac de sang présente 1 litre.
Par exemple, on peut mesurer la pression pour 10 litres de sang, 20 litres, etc. dans notre cas on mesure la pression pour 40 litre de sang dans chaque case.
- ii. Cela nous donne une série de points de données (quantité de sang vs. pression).
- iii. **Relation entre Pression et Quantité de Sang** : On trace une courbe basée sur ces points de données pour établir une relation mathématique (souvent une courbe ou une ligne droite) entre la pression et la quantité de sang. La relation peut être une équation linéaire simple ou une équation plus complexe selon les données obtenues.

Supposons que nous obtenions dans notre cas une relation simple comme [15] :

$$\text{Quantité de sang (litres)} = \frac{\text{Pression (kPa)}}{k} \quad \text{où } k \text{ est une constante obtenue par l'étalonnage} \tag{3.1}$$

Equation 1 : Quantité de sang en fonction de la pression 3.1

(b) **Calcul du pourcentage de capacité**

Pour calculer le pourcentage de capacité de la case [15] :

- **Formule du Pourcentage** :

$$\text{Pourcentage de sang} = \left(\frac{\text{Quantité de sang (litres)}}{\text{Capacité maximale (litres)}} \right) \times 100 \tag{3.2}$$

Equation 2 : Calcul du pourcentage de capacité de la case 3.2

- **Exemple de Calcul** : dans notre cas la capacité maximale par exemple est de 80 litres et que la quantité mesurée est de 25 litres :

$$\text{Pourcentage de sang} = \left(\frac{25 \text{ litres}}{80 \text{ litres}} \right) \times 100 = 31.25\% \tag{3.3}$$

Equation 3 : Exemple de calcul du pourcentage de sang 3.3

3.5.1.2 Capteur SI7021

SI7021 est connecté au ESP32 par l'interface I2C, utilisant LoRa ou WiFi.

Principe de fonctionnement du capteur SI7021[119] :

1. Structure du Capteur SI7021 :

- (a) Capteur d'humidité : Le SI7021 utilise un élément capacitif sensible à l'humidité. L'élément contient un polymère dont la permittivité change en fonction de l'humidité relative de l'air. Ce changement est mesuré et converti en une valeur d'humidité relative.
 - Détection d'humidité :
Conversion Analogique-Digitale : La capacité est mesurée électroniquement et convertie en un signal numérique par un convertisseur analogique-numérique (ADC) intégré.
- (b) Capteur de température : Le capteur de température en silicium mesure la température en utilisant les caractéristiques de conduction de la diode ou du transistor.
 - Détection de température :
Conversion Analogique-Digitale : La variation de la tension ou du courant due à la température est convertie en une valeur numérique par l'ADC intégré.

2. Lecture des données :

- Interface I2C : SI7021 communique avec le microcontrôleur via une interface I2C d'une configuration maître-esclave. (Inter-Integrated Circuit).
- Transmission de données : Le microcontrôleur envoie une commande au capteur via le bus I2C pour lire les données de température et d'humidité. Le capteur répond avec les données demandées.

3. Traitement des données :

- Conversion des données : Les données brutes lues du capteur sont traitées par le microcontrôleur ESP32 pour obtenir des valeurs utilisables d'humidité relative (en %) et de température (en °C).
- Compensation et calibration : SI7021 est souvent pré-calibré en usine pour compenser les erreurs de mesure. Les données de calibration sont stockées dans le capteur et utilisées pour corriger les lectures.

3.5.1.3 Capteur de tension électrique (ZMPT101B) [36]

Sa sortie Voltage Output (VOUT) est connectée à une entrée analogique Analog-to-Digital Converter (ADC) du ESP32.

Fonctionnement de ZMPT101B :

1. Initialisation de l'ADC :

Configurer l'entrée convertisseur analogique-numérique (ADC) du microcontrôleur (ESP32) avec la sortie analogique du capteur pour lire la valeur de signal analogique du ZMPT101B régulièrement, la convertit en une valeur numérique, et la traite pour déterminer la tension actuelle du réseau.

2. Lecture des données :

- Lire la valeur analogique de la sortie du ZMPT101B.
- Convertir cette valeur analogique en une valeur numérique à l'aide de l'ADC.

3. Calcul de la tension AC :

- Convertir la valeur numérique de l'ADC en une tension alternative (AC) en utilisant la courbe d'étalonnage du module.
- Calculer la valeur quadratique moyenne en anglais Root Mean Square (RMS) pour obtenir la mesure de la tension AC.

4. Détection de coupure de courant :

- Le microcontrôleur compare la tension mesurée à un seuil prédéfini dans cette architecture on met 0V pour détecter une coupure de courant.
- Si la tension mesurée tombe en dessous de ce seuil (0V pour une coupure totale), le microcontrôleur reconnaît qu'il y a une coupure de courant ou une anomalie.

5. Activation de l'UPS : un Uninterruptible Power Supply (UPS), ou alimentation sans interruption, est un dispositif électrique qui fournit une alimentation d'urgence à une charge lorsqu'une source d'alimentation principale ou d'entrée échoue. Un UPS est essentiel pour toute infrastructure nécessitant une alimentation continue et stable comme notre cas, assurant ainsi la sécurité et la fiabilité des opérations.

- Lorsqu'une coupure de courant est détectée, le microcontrôleur envoie un signal pour activer l'alimentation sans interruption(Uninterruptible Power Supply ou UPS).
- L'UPS est conçu pour surveiller ce signal et passer instantanément à l'alimentation par batterie dès qu'il reçoit le signal d'activation.

3.5.2 Transmission vers le Fog Computing (Raspberry Pi)

Le microcontrôleur ESP32 lit les données des capteurs et les transmet au dispositif Fog (Raspberry Pi) via la communication LoRa en utilisant les protocoles Message Queuing Telemetry Transport (MQTT) ou HyperText Transfer Protocol (http).

3.5.3 Traitement des données par le Fog Computing

Le Raspberry Pi effectue le traitement des données en trois étapes :

1. Agréger : Le Fog Computing collecte les données provenant des différents capteurs via l'ESP32.
2. Analyser : Le fog Computing exécute un algorithme simple pour vérifier les seuils critiques :
 - ▷ Si la quantité de sang atteint 60%, une appelle est envoyée pour demander des dons de sang.
 - ▷ Si la quantité de sang tombe à 40%, une appelle d'urgence est envoyée.
 - ▷ Si la température n'est pas inférieure à 4°C ou si l'humidité n'est pas inférieure ou égale à 60%, une alerte est envoyée au responsable médical et la LED rouge s'allume.[32]
 - ▷ Si une coupure de courant est détectée par le capteur de tension, le système UPS est activé pour fournir une alimentation de secours.

Algorithm 1 Surveillance du stockage de sang

```

1: function ANALYSER()
2:   // Initialisation des types de sang pour chaque case de chaque étage
3:   TypesSang ←
["A+", "A-" , // Premier étage
"B+", "B-" , // Deuxième étage
"O+", "O-" , // Troisième étage
"AB+", "AB-" // Quatrième étage ]
4:   read temperature-chambre, humidite-chambre, tensionelectrique-chambre
5:   // Vérifier la température et l'humidité
6:   if temperature-chambre ≥ 4.0 or humidite-chambre > 60.0 then
7:     envoyerAlerte("Alerte au responsable médical pour le type de sang " + TypeSang)
8:     allumerLED("rouge")
9:   end if
10:  // Vérifier la tension électrique
11:  if tensionelectrique-chambre == 0 then
12:    activerUPS()
13:    allumerLED("rouge")
14:  end if
15:  for int j = 0 ; j < 4 ; j++ do
16:    for int i = 0 ; i < 8 ; i++ do
17:      // Lire les données des capteurs pour la case i de l'étage j
18:      read quantite_sang pour case [i] de l'étage [j]
19:      int quantite ← quantite_sang[j][i]
20:      String TypeSang ← TypesSang[j][i]
21:      // Vérifier la quantité de sang
22:      if quantite ≤ 60 AND quantite > 40 then
23:        envoyerSMS_via_firebase("Demander des dons de sang pour le type de
sang " + TypeSang)
24:      end if
25:      if quantite ≤ 40 then
26:        envoyerSMS_via_twilio("Appel d'urgence pour don de sang pour le type
de sang " + TypeSang)
27:      end if
28:    end for
29:  end for
30: end function

```

3. Stocker : Les données traitées peuvent être temporairement stockées dans une base de données locale pour permettre des réactions en temps réel et de prendre des décisions immédiate par exemple allumer la LED rouge pour indiquer un problème environnemental.

3.5.4 Transmission vers le Cloud

Après traitement, le fog Computing envoie les données vers une passerelle Long Range Wide Area Network (LoRaWAN), qui les transmet ensuite vers le cloud. Dans le cloud, deux plateformes sont utilisées :

- TWILIO : Pour envoyer des SMS en cas d'urgence.
- Firebase : Pour la gestion des cas normaux et le stockage des données.

3.6 Algorithme globale de gestion du stockage de sang

L'algorithme est conçu pour surveiller et gérer les conditions du stockage du sang dans une chambre spécialisée, en utilisant divers capteurs pour suivre les paramètres critiques tels que la température, l'humidité, la quantité de sang et la tension électrique.

Algorithm 2 Système de surveillance et gestion du stockage de sang

```

1: // Initialisation des types de sang pour chaque case de chaque étage
2: TypesSang ← [ ["A+", "A-"], // Premier étage ["B+", "B-"], // Deuxième étage ["O+",
  "O-"], // Troisième étage ["AB+", "AB-"] // Quatrième étage ]
3: // Constantes pour les seuils critiques
4: constante double SEUIL_TEMPERATURE ← 4.0 // en °C
5: constante double SEUIL_HUMIDITE ← 60.0 // en %
6: constante int SEUIL_QUANTITE_SANG ← 40 // en %
7: constante int SEUIL_TENSION_ELECTRIQUE ← 0 // en volts
8: // Données de la chambre du stockage
9: private static double temperatureChambre
10: private static double humiditeChambre
11: private static int tensionElectriqueChambre
12: // Initialisation des capteurs et lecture des données
13: CapteurPression ← Honeywell26PC
14: CapteurHumidite ← SI7021
15: CapteurTemperature ← SI7021
16: CapteurTensionElectrique ← ZMPT101B
17: // Lire la température, l'humidité et vérifier le courant électrique de la chambre
18: read temperatureChambre depuis SI7021
19: read humiditeChambre depuis SI7021
20: read tensionElectriqueChambre depuis Honeywell26PC
21: // Vérification des conditions et activation de l'UPS
22: if temperatureChambre ≥ 4.0 or humiditeChambre > 60.0 then
23:   envoyerAlerte("Alerte au responsable médical une condition critique")
24:   allumerLEDCritique(temperatureChambre, humiditeChambre, tensionElectrique-
  Chambre)
25: end if
26: function ACTIVERUPS(double tension)
27:   write("system UPS est activé")
28: end function
29: // Vérifier la tension électrique
30: if tensionElectriqueChambre == 0 then
31:   activerUPS(tensionElectriqueChambre)
32: end if
33: // Lire les données de capteur de pression pour chaque case de chaque étage
34: for int j = 0; j < 4; j++ do                                ▷ Boucle pour les étages
35:   for int i = 0; i < 8; i++ do                                ▷ Boucle pour les cases
36:     read quantite_sang depuis ZMPT101B pour la case i de l'étage j
37:     // Déterminer le type de sang pour la case i de l'étage j
38:     String TypeSang ← TypesSang[j][i]
39:     // Vérifier la quantité de sang
40:     if quantite_sang[j][i] ≤ 60 AND quantite_sang[j][i] > 40 then
41:       envoyerSMS via firebase("Demander des dons de sang pour le type de sang "
  + TypeSang + " mise à jour de la carte")
42:     end if
43:     if quantite_sang[j][i] ≤ 40 then
44:       envoyerSMS via twilio("Appel d'urgence pour don de sang pour le type de sang
  " + TypeSang + " mise à jour de la carte")
45:     end if
46:   end for
47: end for

```

Algorithm 3 La suite de l'algorithme système de surveillance et gestion du stockage de sang

```
1: // Connexion du microcontrôleur et transmission des données
2: Microcontroleur ← ESP32
3: Relier CapteurPression, CapteurHumidite, CapteurTemperature, CapteurTensionElec-
   trique au Microcontroleur
4: // Transmission des données vers le Fog (Raspberry Pi)
5: FOG ← RaspberryPi
6: Microcontroleur.EnvoyerDonnees(FOG, "LoRa", "MQTT")
7: // Traitement des données par le Fog Computing et Transmission au Cloud
8: FOG.Agréger()
9: FOG.Analyser()
10: FOG.StockerDonnéesLocalement()
11: // Envoyer les informations au cloud via LoRaWAN
12: PasserelleLoRaWAN ← Transmettre(SystemeFOG)
13: Cloud ← EnvoyerDonnées(PasserelleLoRaWAN)
14: // Interface utilisateur pour les notifications
15: InterfaceUtilisateur ← HebergerInterface(Cloud)
16: InterfaceUtilisateur.EnvoyerSMS("Demander des dons de sang via Firebase ou Twilio en
   cas d'urgence")
17: function ALLUMERLEDCRITIQUE(double temperature, double humidite, int quan-
   tite_sang, int tension_electrique)
18:   if temperature != 4.0 or humidite > 60.0 or quantite_sang ≤ 40 or ten-
   sion_electrique == 0 then
19:     Allumer LED rouge
20:   end if
21: end function
22: function ALLUMERLEDCRITIQUE(double temperature, double humidite, int tensionE-
   lectrique)
23:   if temperature ≥ 4.0 or humidite > 60.0 or tensionElectrique == 0 then
24:     write("Conditions critiques détectées : Allumage de la LED rouge")
25:   else
26:     write("Conditions normales, la LED reste éteinte")
27:   end if
28: end function
```

3.6.1 Explication de l’algorithme

1. Initialisation des types de sang et des seuils critiques : L’algorithme commence par définir les types de sang stockés à chaque étage de la chambre du stockage. Chaque étage est assigné à deux types de sang. Ensuite, il définit des constantes pour les seuils critiques de température (4°C), d’humidité (60%), de quantité de sang (40%) et de tension électrique (0 volts). Ces seuils sont utilisés pour déterminer si les conditions du stockage sont adéquates.

Algorithm 4 Initialisation des types de sang et des seuils critiques

```

1: function INITIALISER
2:   // Initialisation des types de sang pour chaque case de chaque étage
3:   TypesSang ← [
4:     ["A+", "A-"], // Premier étage
5:     ["B+", "B-"], // Deuxième étage
6:     ["O+", "O-"], // Troisième étage
7:     ["AB+", "AB-"] // Quatrième étage
8:   ]
9:   // Constantes pour les seuils critiques
10:  constante double SEUIL_TEMPERATURE ← 4.0 // en °C
11:  constante double SEUIL_HUMIDITE ← 60.0 // en %
12:  constante int SEUIL_QUANTITE_SANG ← 40 // en %
13:  constante int SEUIL_TENSION_ELECTRIQUE ← 0 // en volts
14: end function

```

2. Déclaration des variables de la chambre de stockage : Des variables sont déclarées pour stocker les valeurs de température, d’humidité et de tension électrique lues à partir des capteurs. Ces variables sont essentielles pour surveiller en temps réel les conditions de la chambre du stockage.

Algorithm 5 Données de la chambre de stockage

```

1: private static double temperatureChambre
2: private static double humiditeChambre
3: private static int tensionElectriqueChambre

```

3. Initialisation des capteurs : Les capteurs utilisés dans le système sont initialisés, notamment les capteurs de pression, d’humidité, de température et de tension électrique. Chaque capteur est associé à un modèle spécifique, ce qui permet de lire les données environnementales nécessaires.

Algorithm 6 Initialisation des capteurs et lecture des données

```
1: // Initialisation des capteurs
2: CapteurPression ← Honeywell26PC
3: CapteurHumidite ← SI7021
4: CapteurTemperature ← SI7021
5: CapteurTensionElectrique ← ZMPT101B
6: // Lire la température, l'humidité et vérifier le courant électrique de la chambre
7: read temperatureChambre depuis SI7021
8: read humiditeChambre depuis SI7021
9: read tensionElectriqueChambre depuis Honeywell26PC
```

4. Garantir la sécurité des stocks de sang grâce à une surveillance rigoureuse de la température et de l'humidité : L'algorithme met en œuvre un système de surveillance robuste pour maintenir des conditions optimales dans la chambre de stockage. Il compare en permanence les valeurs mesurées aux seuils critiques et déclenche des actions correctives en cas de déviation. Les alertes immédiates et les signaux visuels permettent au personnel médical de réagir rapidement, assurant ainsi la préservation des stocks de sang précieux.

Algorithm 7 Vérification des conditions et activation de l'UPS

```
1: // Vérifier la température et l'humidité
2: if temperatureChambre ≥ 4.0 or humiditeChambre > 60.0 then
3:   envoyerAlerte("Alerte au responsable médical une condition critique")
4:   allumerLEDCritique(temperatureChambre, humiditeChambre, tensionElectrique-
   Chambre)
5: end if
6: function ACTIVERUPS(double tension)
7:   write("system UPS est activé")
8: end function
9: // Vérifier la tension électrique
10: if tensionElectriqueChambre == 0 then
11:   activerUPS(tensionElectriqueChambre)
12: end if
```

5. Lecture et vérification de la quantité de sang dans chaque case de chaque étage : L'algorithme parcourt chaque étage et chaque case pour lire la quantité de sang disponible. Il vérifie ensuite si la quantité de sang dépasse 60% ou tombe en dessous de 40%. En fonction de ces vérifications, des messages SMS sont envoyés via Firebase ou Twilio pour demander des dons de sang ou lancer un appel d'urgence.

Algorithm 8 Lire les données de capteur de pression dans chaque case de chaque étage

```
1: for int j = 0; j < 4; j++ do                                ▷ Boucle pour les étages
2:   for int i = 0; i < 8; i++ do                              ▷ Boucle pour les cases
3:     read quantite_sang depuis ZMPT101B pour la case i de l'étage j
4:     // Déterminer le type de sang pour la case i de l'étage j
5:     String TypeSang ← TypesSang[j][i]
6:     // Vérifier la quantité de sang
7:     if quantite_sang[j][i] ≤ 60 AND quantite_sang[j][i] > 40 then
8:       envoyerSMS via firebase("Demander des dons de sang pour le type de sang "
+ TypeSang + " mise à jour de la carte")
9:     end if
10:    if quantite_sang[j][i] ≤ 40 then
11:      envoyerSMS via twilio("Appel d'urgence pour don de sang pour le type de sang
" + TypeSang + " mise à jour de la carte")
12:    end if
13:  end for
14: end for
```

6. Connexion des capteurs au microcontrôleur et transmission des données : Les capteurs sont connectés à un microcontrôleur (ESP32), qui envoie les données au Fog (Raspberry Pi) en utilisant les protocoles LoRa et MQTT. Cette connexion permet une surveillance continue et la transmission des données en temps réel.

Algorithm 9 Connexion du microcontrôleur et transmission des données

```
1: // Connecter le microcontrôleur
2: Microcontroleur ← ESP32
3: Relier CapteurPression, CapteurHumidite, CapteurTemperature, CapteurTensionElec-
   trique au Microcontroleur
4: // Transmission des données vers le Fog (Raspberry Pi)
5: FOG ← RaspberryPi
6: Microcontroleur.EnvoyerDonnees(FOG, "LoRa", "MQTT")
```

7. Traitement des données par Fog et transmission au cloud : Fog agrège, analyse et stocke les données localement. Ensuite, les informations sont envoyées au cloud via une passerelle LoRaWAN, permettant un accès distant et sécurisé aux données de la chambre de stockage.

Algorithm 10 Traitement des Données par le Fog et Transmission au Cloud

```
1: // Traitement des Données par le Fog
2: FOG.Agréger()
3: FOG.Analyser()
4: FOG.StockerDonnéesLocalement()
5: // Envoyer les informations au cloud via LoRaWAN
6: PasserelleLoRaWAN ← Transmettre(FOG)
7: Cloud ← EnvoyerDonnées(PasserelleLoRaWAN)
```

8. Interface utilisateur et notifications : Une interface utilisateur est hébergée dans le cloud pour gérer les notifications. Cette interface envoie des SMS pour demander des dons de sang en cas d'urgence, assurant ainsi une réponse rapide aux besoins critiques.

Algorithm 11 Interface utilisateur pour les notifications

```
1: // Interface utilisateur pour les notifications
2: InterfaceUtilisateur ← HebergerInterface(Cloud)
3: InterfaceUtilisateur.EnvoyerSMS("Demander des dons de sang via Firebase ou Twilio en cas d'urgence")
```

9. Fonction pour allumer la LED en cas de conditions critiques : Une fonction spécifique allume une LED rouge en cas de détection de conditions critiques (température, humidité, quantité de sang ou tension électrique anormales). Cette indication visuelle aide le personnel à réagir rapidement en cas de problème. Par exemple, un employé qui passe dans le couloir de la banque de stockage de sang peut savoir qu'il y a un problème dans la chambre de stockage même s'il n'y accède pas.

Algorithm 12 Fonction d'allumage de la LED critique

```
1: function ALLUMERLEDCRITIQUE(double temperature, double humidite, int tensionE-
   lectrique)
2:   if temperature ≥ 4.0 or humidite > 60.0 or tensionElectrique == 0 then
3:     write("Conditions critiques détectées : Allumage de la LED rouge")
4:   else
5:     write("Conditions normales, la LED reste éteinte")
6:   end if
7: end function
```

3.7 Partie pratique et simulation

3.7.1 Environnement de programmation (Google Colab)

un service gratuit de Google qui permet d'exécuter du code Python dans un navigateur web. Il est particulièrement utile pour le machine learning, l'analyse de données et le développement collaboratif. Colab offre des GPU gratuits pour l'accélération de calculs, l'intégration avec Google Drive pour le stockage des fichiers, et la possibilité de partager facilement des notebooks avec d'autres utilisateurs.[80]

3.7.2 Algorithme Python globale de surveillance et de gestion des stocks de sang

Le programme vise à simuler un système de surveillance et de gestion des stocks de sang en générant des données de capteurs fictives. Ces données incluent la pression (quantité de sang), la température, l'humidité et la tension électrique. Le programme analyse ces données, déclenche des notifications basées sur des seuils prédéfinis, et visualise les résultats à travers divers graphiques.

```

1 import random # Instruction permettant de generer des nombres aleatoires
2 import pandas as pd # Pour la manipulation et l'analyse des donnees
3 import matplotlib.pyplot as plt # Pour creer des visualisations graphiques
4 import seaborn as sns # Pour la creation de visualisations statistiques
5 import time # Pour la mesure du temps d'execution et la gestion des delais
6
7 # Definir les types de sang disponibles pour la gestion du stock
8 types_sang = ["A+", "A-", "B+", "B-", "O+", "O-", "AB+", "AB-"]
9
10 # Definir les seuils pour les notifications
11 SEUIL_ALERT = 4.0 # Seuil de temperature critique
12 SEUIL_HUMIDITE = 60.0 # Seuil d'humidite critique
13 SEUIL_SANG_NORMAL = 60 # Seuil normal de quantite de sang
14 SEUIL_SANG_URGENCE = 40 # Seuil d'urgence de quantite de sang critique
15
16 # Generer des donnees aleatoires pour simuler les capteurs
17 def generate_random_data(num_records):
18     data = [] # Liste qui stocke les donn es g n r es
19     notifications_alerte_manager = 0
20     notifications_demande_normale_manager = 0
21     notifications_demande_urgence_manager = 0
22     notifications_envoyees_cloud_normales = 0
23     notifications_envoyees_cloud_urgentes = 0
24     notifications_envoyees_cloud_par_type = {type_sang: {"normales": 0, "
urgentes": 0} for type_sang in types_sang}
25
26     while len(data) < num_records: # Correction de la condition pour
g n r e r le nombre souhait de donn es
27         for type_sang in types_sang:
28             if len(data) >= num_records:
29                 break
30
31         # Generation des donnees pour les capteurs de pression
32         quantite_sang = random.randint(1, 100) # Valeurs entre 1% et 100%
33         temp = f"2024-07-01T{random.randint(0, 23):02d}:00:00"

```

```

34     led_state = quantite_sang <= SEUIL_SANG_NORMAL
35     ups_state = False # Initialiser UPS comme inactif par d faut
36
37     # Gestion des notifications pour le manager
38     if quantite_sang <= SEUIL_SANG_NORMAL:
39         notifications_demande_normale_manager += 1
40         notifications_envoyees_cloud_normales += 1
41         notifications_envoyees_cloud_par_type[type_sang][ "normales" ]
+= 1
42     if quantite_sang <= SEUIL_SANG_URGENCE:
43         notifications_demande_urgence_manager += 1
44         notifications_envoyees_cloud_urgentes += 1
45         notifications_envoyees_cloud_par_type[type_sang][ "urgentes" ]
+= 1
46
47     data.append([ len(data) + 1, temp, "Pression", type_sang,
quantite_sang, led_state, ups_state])
48
49     if len(data) >= num_records:
50         break
51
52     # Generation des donnees pour les autres capteurs
53     for capteur in ["Temperature", "Humidite", "Tension"]:
54         temp = f"2024-06-24T{random.randint(0, 23):02d}:00:00"
55         if capteur == "Temperature":
56             valeur = -10 + random.random() * 50 # Valeurs entre -10
C et 40 C
57             led_state = valeur >= SEUIL_ALERT # LED allum e si la
temp rature est sup rieur e ou gale 4 C
58             elif capteur == "Humidite":
59                 valeur = 10 + random.random() * 90 # Valeurs entre 10% et
100%
60                 led_state = valeur != SEUIL_HUMIDITE # LED allum e si l'
humidit e est diff rente de 60%
61             elif capteur == "Tension":
62                 valeur = random.randint(0, 1) # generer aleatoirement 0
ou 1 pour la tension
63                 led_state = valeur == 0 # LED allumee si la tension est
egale zero
64
65                 if valeur == 0: # Verifier si la tension est egale
zero
66                     ups_state = True # UPS actif si la tension est egale
zero
67                 else:
68                     ups_state = False # Sinon, UPS inactif
69
70     # Gestion des notifications d'alerte pour le manager
71     if (capteur == "Temperature" and valeur >= SEUIL_ALERT) or \
72         (capteur == "Humidite" and valeur != SEUIL_HUMIDITE) or \
73         (capteur == "Tension" and valeur == 0):
74         notifications_alerte_manager += 1
75
76     data.append([ len(data) + 1, temp, capteur, "", valeur,
led_state, ups_state])
77
78     if len(data) >= num_records:
79         break
80

```

```

81     return (
82         data[:num_records],
83         notifications_alerte_manager,
84         notifications_demande_normale_manager,
85         notifications_demande_urgence_manager,
86         notifications_envoyees_cloud_normales,
87         notifications_envoyees_cloud_urgentes,
88         notifications_envoyees_cloud_par_type,
89     )
90
91 # Mesurer le temps d'ex cution
92 start_time = time.time()
93
94 # G n rer les donn es
95 (
96     data,
97     notifications_alerte_manager,
98     notifications_demande_normale_manager,
99     notifications_demande_urgence_manager,
100    notifications_envoyees_cloud_normales,
101    notifications_envoyees_cloud_urgentes,
102    notifications_envoyees_cloud_par_type,
103 ) = generate_random_data(800)
104
105 # Convertir en DataFrame pandas
106 df = pd.DataFrame(
107     data,
108     columns=["ID", "Temps", "TypeCapteur", "TypeSang", "Valeur", "LedAllumee",
109             "UpsActive"]
110 )
111 # Afficher les donnees
112 pd.options.display.float_format = '{:,.0f}'.format # Formatage des nombres
113     sans z ros apr s la virgule
114 print(df)
115 # Afficher les statistiques de notifications pour le manager
116 print("\nNotifications pour le manager :")
117 print(f" - Alertes (anomalie temp rature / tension lectrique / humidit ) :
118     {notifications_alerte_manager}")
119 print(f" - Demandes normales de sang : {notifications_demande_normale_manager}
120     ")
121 print(f" - Demandes urgentes de sang : {notifications_demande_urgence_manager}
122     ")
123
124 # Afficher les statistiques de notifications pour le cloud
125 print("\nNotifications envoy es vers le cloud :")
126 print(" - Normales : ")
127 for i, type_sang in enumerate(types_sang):
128     normales = notifications_envoyees_cloud_par_type[type_sang][ "normales" ]
129     print(f"     {i+1}. {type_sang}: {normales}")
130 print(" - Urgentes : ")
131 for i, type_sang in enumerate(types_sang):
132     urgentes = notifications_envoyees_cloud_par_type[type_sang][ "urgentes" ]
133     print(f"     {i+1}. {type_sang}: {urgentes}")
134
135 # Temps d'execution
136 execution_time = time.time() - start_time
137 print(f"\nTemps d'ex cution : {execution_time} secondes")

```

```

135
136 # Tracer les donnees sous forme de graphique
137 def plot_data(df):
138     df_pression = df[df["TypeCapteur"] == "Pression"]
139     df_autres_capteurs = df[
140         (df["TypeCapteur"] == "Temperature")
141         | (df["TypeCapteur"] == "Humidite")
142         | (df["TypeCapteur"] == "Tension")
143     ]
144
145 # Graphique pour la pression par type de sang
146 plt.figure(figsize=(12, 8))
147 sns.barplot(x="TypeSang", y="Valeur", data=df_pression, palette="viridis")
148 plt.title("Quantit de sang par type")
149 plt.xlabel("Type de Sang")
150 plt.ylabel("Quantit (%)")
151 plt.show()
152
153 plt.figure(figsize=(12, 8))
154 sns.lineplot(data=df_pression, x="Temps", y="Valeur", hue="TypeSang",
155 marker="o")
156 plt.title("Quantit de sang par type au fil du temps")
157 plt.xlabel("Temps")
158 plt.ylabel("Quantit (%)")
159 plt.xticks(rotation=45)
160 plt.legend(loc="upper right")
161 plt.show()
162
163 # Graphique pour les autres capteurs (temp rature, humidit , tension)
164 plt.figure(figsize=(12, 8))
165 sns.lineplot(data=df_autres_capteurs, x="Temps", y="Valeur", hue="
166 TypeCapteur", marker="o")
167 plt.title("Donn es des autres capteurs au fil du temps")
168 plt.xlabel("Temps")
169 plt.ylabel("Valeur")
170 plt.xticks(rotation=45)
171 plt.legend(loc="upper right")
172 plt.show()
173
174 # Graphique pour les notifications re ues par le manager (en forme de pie
175 chart)
176 plt.figure(figsize=(10, 8))
177 labels = ['Alertes', 'Demandes normales de sang', 'Demandes urgentes de
178 sang']
179 sizes = [notifications_alerte_manager,
180 notifications_demande_normale_manager,
181 notifications_demande_urgence_manager]
182 plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140, colors=
183 sns.color_palette("viridis", len(labels)))
184 plt.title("Notifications re ues par le manager")
185 plt.show()
186
187 # Graphique pour les notifications envoyees vers le cloud par type de sang
188 (en forme de pie chart)
189 plt.figure(figsize=(10, 8))
190 labels = types_sang
191 normales_percentages = [
192     notifications_envoyees_cloud_par_type[type_sang]["normales"] / sum(
193     notifications_envoyees_cloud_par_type[type_sang].values()) * 100

```

```

185     if sum(notifications_envoyees_cloud_par_type[type_sang].values()) > 0
186     else 0
187     for type_sang in types_sang
188     ]
189     urgentes_percentages = [
190     notifications_envoyees_cloud_par_type[type_sang]["urgentes"] / sum(
191     notifications_envoyees_cloud_par_type[type_sang].values()) * 100
192     if sum(notifications_envoyees_cloud_par_type[type_sang].values()) > 0
193     else 0
194     for type_sang in types_sang
195     ]
196     normales_colors = sns.color_palette("Blues", len(labels))
197     urgentes_colors = sns.color_palette("Reds", len(labels))
198
199     # Plot pour les notifications normales
200     plt.pie(normales_percentages, labels=labels, autopct='%1.1f%%', startangle
201     =140, colors=normales_colors)
202     plt.title('Notifications normales envoyees vers le cloud par type de sang'
203     )
204     plt.show()
205
206     # Plot pour les notifications urgentes
207     plt.figure(figsize=(10, 8))
208     plt.pie(urgentes_percentages, labels=labels, autopct='%1.1f%%', startangle
209     =140, colors=urgentes_colors)
210     plt.title('Notifications urgentes envoyees vers le cloud par type de sang'
211     )
212     plt.show()
213
214     # Tracer les données
215     plot_data(df)

```

3.7.3 Description de programme python

3.7.3.1 Outils de programmation utilisés

- Pandas : Pour la manipulation et l'analyse des données.
- Matplotlib : Pour créer des visualisations graphiques.
- Seaborn : Pour la création de visualisations statistiques.
- Random : Pour générer des nombres aléatoires, simulant les données des capteurs.
- Time : Pour mesurer le temps d'exécution et gérer les délais.

3.7.3.2 Explication des parties principales de programme

1. Importation des bibliothèques :

```
import random
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import time
```

FIGURE 3.8 – Importation des bibliothèques.

- `import random` : instruction offre des fonctions pour générer des nombres aléatoires.
- `import pandas as pd` : importe le module pandas sous l’alias pd. pour la manipulation et l’analyse de données structurées (comme des tableaux ou des bases de données).
- `import matplotlib.pyplot as plt` : importe matplotlib.pyplot sous l’alias plt, permet de créer des graphiques et des visualisations à partir des données manipulées avec Pandas ou d’autres sources de données
- `import seaborn as sns` : bibliothèque de visualisation statistique en Python.
- `import time` : bibliothèque standard de Python utilisée pour mesurer les opérations temporelles.

2. Définition des variables globales : Définir les seuils pour les notifications.

```
types_sang = ["A+", "A-", "B+", "B-", "O+", "O-", "AB+", "AB-"]
SEUIL_ALERT = 4.0
SEUIL_HUMIDITE = 60.0
SEUIL_SANG_NORMAL = 60
SEUIL_SANG_URGENCE = 40
```

FIGURE 3.9 – Définition des variables globales.

3. Génération des données aléatoires

`def generate_random_data(num_records)` : C’est une fonction Python qui prend un paramètre `num_records`, représentant le nombre de données (ou enregistrements) à générer. génère des données aléatoires pour chaque type de sang et capteur. Elle simule les quantités de sang et les valeurs de température, d’humidité et de tension.

```
# Fonction pour générer des données aléatoires pour simuler les capteurs
def generate_random_data(num_records):
    data = [] # Liste qui stocke les données générées
    notifications_alerte_manager = 0
    notifications_demande_normale_manager = 0
    notifications_demande_urgence_manager = 0
    notifications_envoyees_cloud_normales = 0
    notifications_envoyees_cloud_urgentes = 0
    notifications_envoyees_cloud_par_type = {"normales": 0, "urgentes": 0}

    while len(data) < num_records: # Tant que le nombre de données générées est inférieur
        for type_sang in types_sang:
            if len(data) >= num_records: # Vérifie si le nombre de données atteint le nom
                break # Sort de la boucle for si assez de données ont été générées
```

FIGURE 3.10 – Génération des données aléatoires.

4. Mesure du temps d'exécution : La fonction `time.time()` est utilisée pour mesurer le temps total d'exécution du programme.

```
start_time = time.time()
```

FIGURE 3.11 – Mesure du temps d'exécution.

5. Conversion en DataFrame Pandas : Les données générées sont converties en un DataFrame Pandas pour une manipulation et une analyse plus aisées.

```
df = pd.DataFrame(
    data,
    columns=["ID", "Temps", "TypeCapteur", "TypeSang", "Valeur", "LedAllumee", "UpsAct
    )
```

FIGURE 3.12 – Conversion en DataFrame Pandas.

6. Affichage des données et statistiques : Les données et les statistiques des notifications sont affichées à l'écran.

```
print(df)
# Afficher les statistiques de notifications pour le manager et le cloud...
```

FIGURE 3.13 – Affichage des données et statistiques.

7. Tracé des données : La fonction `plot_data(df)` est utilisée pour créer des visualisations graphiques des données.

```
def plot_data(df):
    # Code pour tracer les graphiques...
```

FIGURE 3.14 – Tracé des donnée.

8. Exécution de la fonction de tracé : Les graphiques sont générés et affichés.

```
plot_data(df)
```

FIGURE 3.15 – Exécution de la fonction de tracé.

3.7.4 Exécution de l’algorithme Python dans Google Colab

3.7.4.1 Dataset

Ce dataset représente des données simulées de capteurs collectées sur une période du temps spécifique. Chaque enregistrement contient les informations suivantes :

- **ID** : Identifiant unique de l’enregistrement.
- **Temps** : Timestamp indiquant le moment de la mesure (2024-07-01T08 :30 :00)¹.
- **TypeCapteur** : Type de capteur utilisé pour la mesure (pression, température, humidité, tension).
- **TypeSang** : Type de sang associé à la mesure.
- **Valeur** : Valeur mesurée par le capteur, spécifique au type de capteur.
- **LedAllumee** : Indicateur booléen indiquant si la LED est allumée (True) ou éteinte (False) en fonction de certaines conditions.
- **UpsActive** : Indicateur booléen indiquant si l’UPS (alimentation sans coupure) est active (True) ou non (False), principalement influencé par la mesure de la tension.

1. Chaîne de caractères "2024-07-01T08 :30 :00" le 01 juillet 2024 à 08 heures, 30 minutes et 00 secondes. Cette forme est couramment utilisée dans les formats de date et heure normalisés comme spécifié par la norme ISO 8601, ce qui facilite le traitement et la comparaison des dates dans les systèmes informatiques[58].

stockagedesang.ipynb

Fichier Modifier Affichage Insérer Exécution Outils Aide Toutes les modifications ont été enregistrées

+ Code + Texte

ID	Temps	TypeCapteur	TypeSang	Valeur	LedAllumee	\
0	1 2024-07-01T04:00:00	Pression	A+	86	False	
1	2 2024-06-24T01:00:00	Temperature		6	True	
2	3 2024-06-24T05:00:00	Humidite		66	True	
3	4 2024-06-24T03:00:00	Tension		0	True	
4	5 2024-07-01T17:00:00	Pression	A-	32	True	
..
795	796 2024-06-24T16:00:00	Tension		1	False	
796	797 2024-07-01T12:00:00	Pression	AB-	48	True	
797	798 2024-06-24T18:00:00	Temperature		15	True	
798	799 2024-06-24T14:00:00	Humidite		93	True	
799	800 2024-06-24T10:00:00	Tension		1	False	

UpsActive

0	False
1	False
2	False
3	True
4	False
..	...
795	False
796	False
797	False
798	False
799	False

[800 rows x 7 columns]

11 s terminée à 09:59

FIGURE 3.16 – Dataset utilisé.

3.7.4.2 Présentation des notifications envoyées

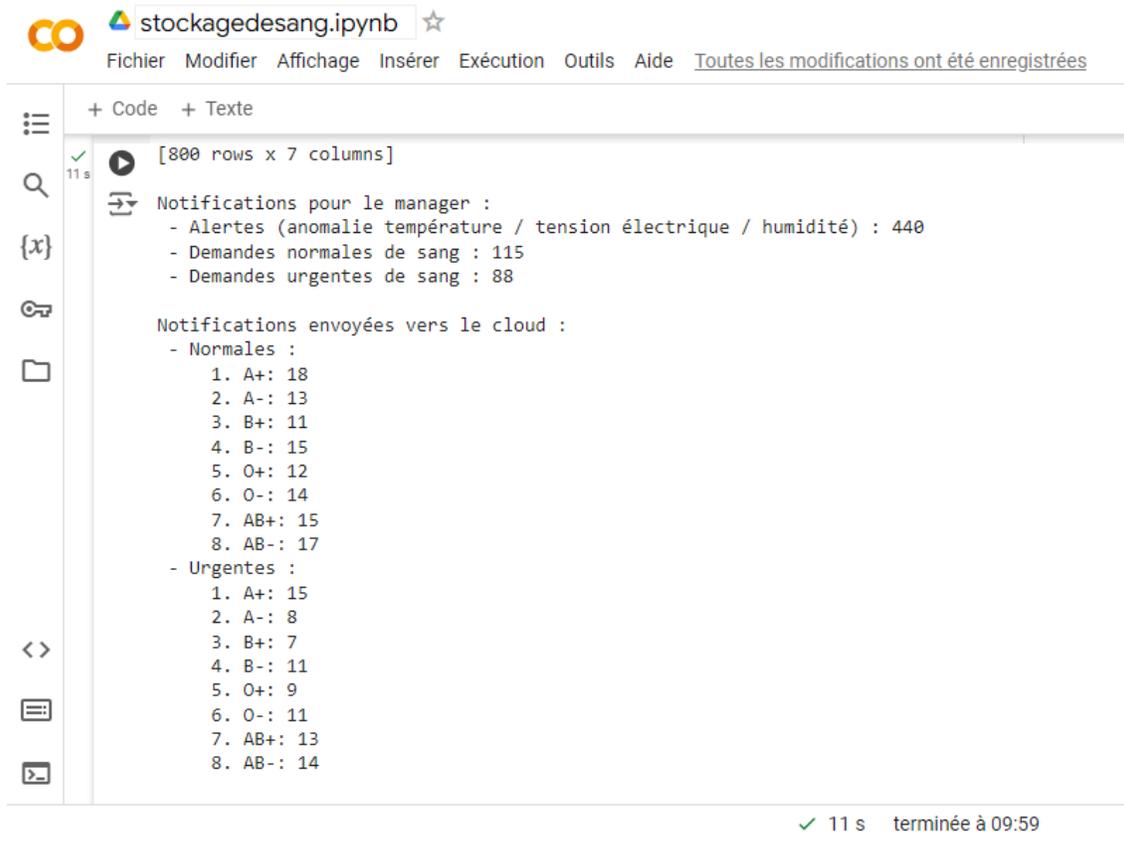
Cette partie fournit un résumé des notifications générées par le système de capteurs et leur traitement :

1. Notifications pour le manager :

- Alertes (température, tension électrique, humidité) : 449 alertes ont été envoyées au manager en raison de valeurs anormales détectées par les capteurs.
- Demandes normales de sang : 84 notifications ont été envoyées pour des niveaux de sang critiques mais non urgents.
- Demandes urgentes de sang : 39 notifications ont été envoyées en raison de niveaux de sang critiques nécessitant une action immédiate.

2. Notifications envoyées vers le cloud :

- Normales : Notifications standard envoyées pour surveiller les niveaux de sang qui ne sont pas critiques.
- Urgentes : Notifications critiques envoyées pour des niveaux de sang nécessitant une intervention immédiate.



```
stockagedesang.ipynb
Fichier Modifier Affichage Insérer Exécution Outils Aide Toutes les modifications ont été enregistrées

+ Code + Texte

[800 rows x 7 columns]
11 s
Notifications pour le manager :
- Alertes (anomalie température / tension électrique / humidité) : 440
- Demandes normales de sang : 115
- Demandes urgentes de sang : 88

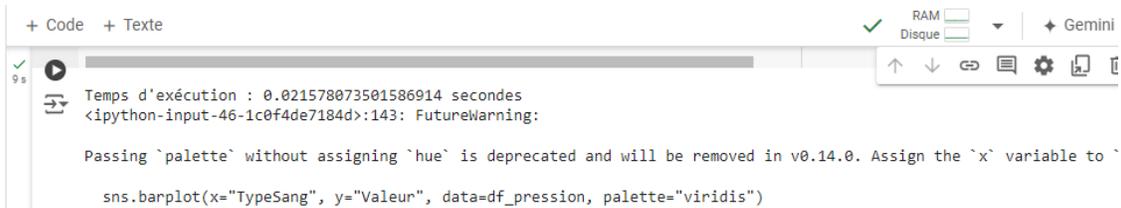
Notifications envoyées vers le cloud :
- Normales :
  1. A+: 18
  2. A-: 13
  3. B+: 11
  4. B-: 15
  5. O+: 12
  6. O-: 14
  7. AB+: 15
  8. AB-: 17
- Urgentes :
  1. A+: 15
  2. A-: 8
  3. B+: 7
  4. B-: 11
  5. O+: 9
  6. O-: 11
  7. AB+: 13
  8. AB-: 14

✓ 11 s terminée à 09:59
```

FIGURE 3.17 – Notifications envoyées.

3. Durée d'exécution :

Temps d'exécution : 0.01798391342163086 secondes ont été nécessaires pour exécuter l'ensemble du programme.



```
+ Code + Texte
RAM Disque Gemini
9 s
Temps d'exécution : 0.021578073501586914 secondes
<ipython-input-46-1c0f4de7184d>:143: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `
sns.barplot(x="TypeSang", y="Valeur", data=df_pression, palette="viridis")
```

FIGURE 3.18 – Temps d'exécution.

3.7.4.3 Graphiques de présentation

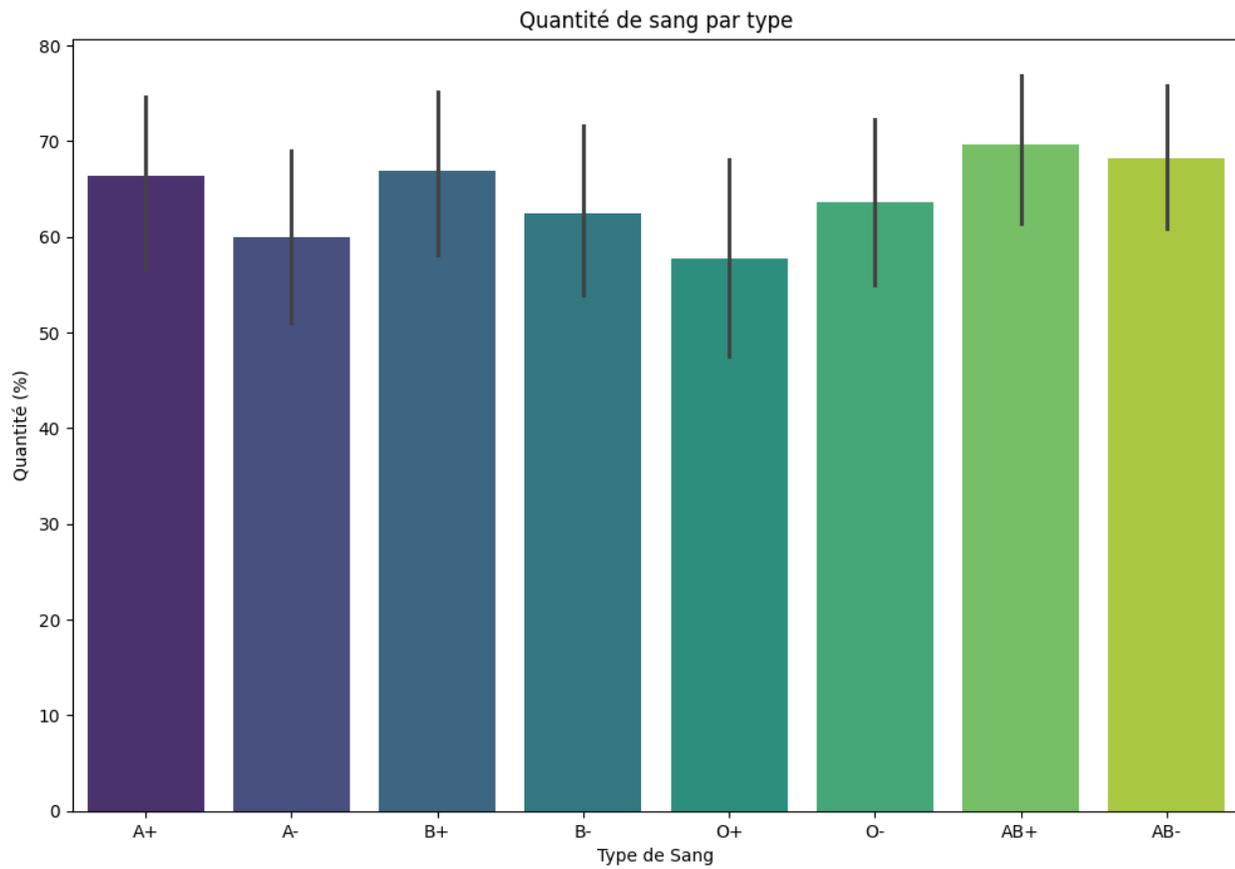


FIGURE 3.19 – Diagrammes à barres illustrant la quantité de chaque type de sang[79].

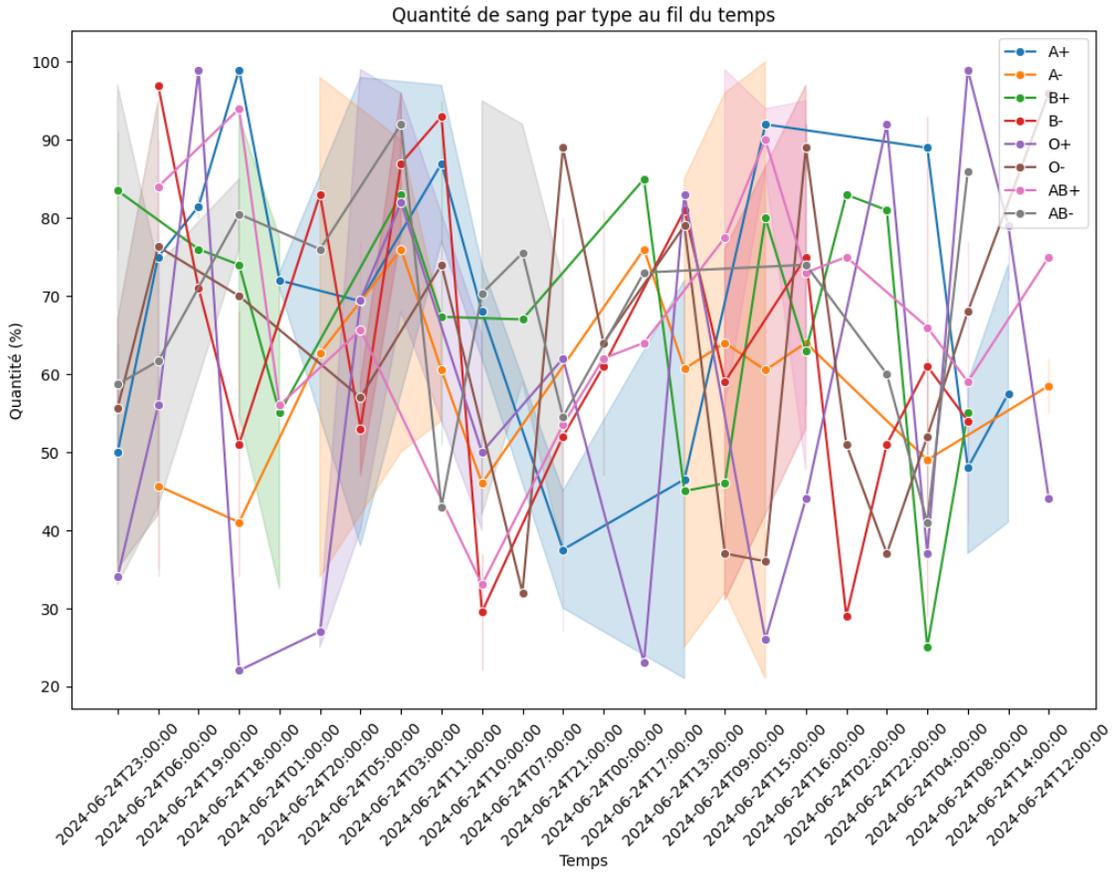


FIGURE 3.20 – Évolution temporelle de la quantité de chaque type de sang en courbes[79].

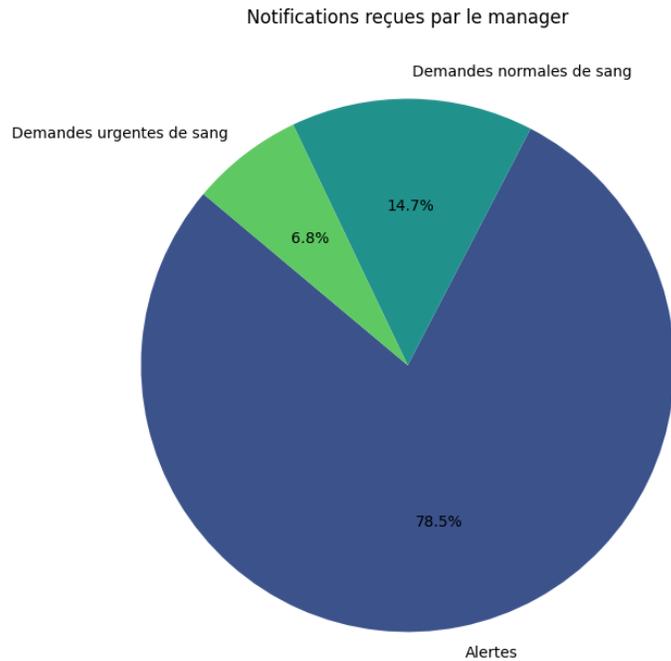


FIGURE 3.21 – Présentation sous forme de diagramme circulaire des notifications reçues par le manager[79].

Notifications normales envoyées vers le cloud par type de sang

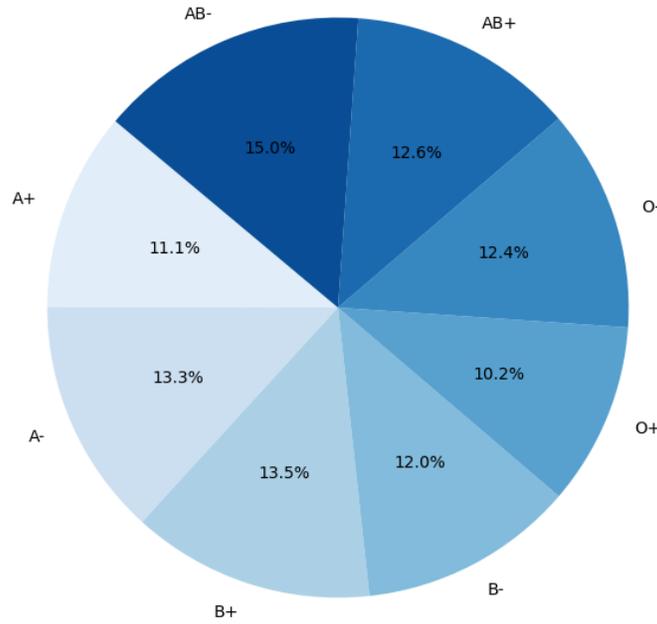


FIGURE 3.22 – Diagramme circulaire représentant les notifications normales vers le cloud selon le type de sang[79].

Notifications urgentes envoyées vers le cloud par type de sang

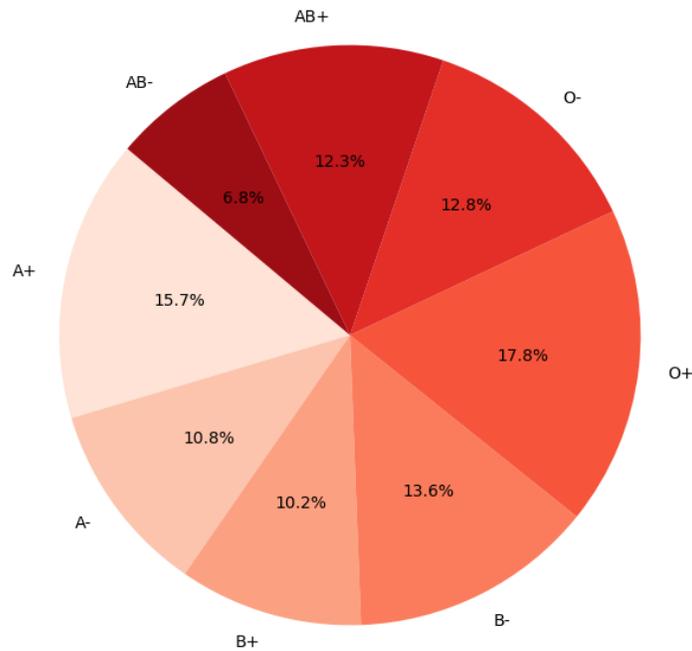


FIGURE 3.23 – Diagramme circulaire représentant les notifications urgentes vers le cloud selon le type de sang[79].

3.8 Carte de fidélité de sang - Avantages et fonctionnalités

Pour encourager les donateurs de sang à participer régulièrement, nous proposons la création d'une "carte de fidélité de sang". Cette carte sera délivrée à chaque personne qui donne son sang. À chaque don, la carte sera mise à jour et activée.

3.8.1 Avantages de la carte

Les avantages de cette carte sont nombreux :

- Réduction sur le prix des consultations médicales en cas de demande de sang normale .
- Obtention d'un rendez-vous médical gratuit en cas d'urgence.
- Possibilité de transférer ces avantages à un membre de la famille.

3.8.2 Caractéristiques de la carte

Les caractéristiques de la carte de fidélité de sang incluront :

- Le groupe sanguin.
- La date d'activation (renouvelée tous les trois mois après chaque don de sang).
- Une date d'expiration.
- La désactivation après chaque utilisation pour une consultation.
- Les coordonnées personnelles du titulaire pour une gestion optimale.

3.9 Interface de don de sang connectée

Une plateforme web connectée au cloud Firebase est développée pour l'architecture IoT mise en place pour la gestion automatisée du stockage de sang. Cette plateforme centralise les données issues des capteurs et des donneurs, offrant ainsi une interface intuitive et sécurisée pour une gestion optimale des ressources sanguines.

3.9.1 Objectifs principaux de la plateforme web

La plateforme web offre aux donneurs :

- Un espace sécurisé pour gérer leurs dons de sang.
- Ils peuvent s'inscrire et se connecter en toute sécurité.
- Consulter leur historique de dons.
- Planifier de nouveaux rendez-vous.
- Recevoir des notifications personnalisées sur les besoins urgents.
- Ils restent informés des dernières actualités et événements liés au don du sang

3.9.2 Firebase : services principaux utilisés

1. Realtime Database

La Realtime Database [30] de Firebase est un service de base de données NoSQL cloud en temps réel. Elle est utilisée pour stocker les données des capteurs en temps réel et les informations dynamiques des utilisateurs. Cette base de données est particulièrement adaptée pour les applications nécessitant des mises à jour en temps réel et une synchronisation instantanée entre les utilisateurs connectés. Les données des capteurs, telles que les niveaux de pression sanguine, les conditions de température et d'humidité, sont stockées ici pour une visualisation et une analyse rapides.

2. Firestore

Firestore[30] est une base de données NoSQL flexible et évolutive de Firebase, utilisée pour stocker les données structurées des utilisateurs inscrits. Elle est conçue pour fournir une synchronisation en temps réel et une haute disponibilité des données, permettant une gestion efficace des profils d'utilisateurs, de leur historique de dons, des rendez-vous planifiés, etc. Cette solution est optimale pour les applications nécessitant des opérations de lecture/écriture fréquentes sur des données structurées.

3. Authentication

Le service d'authentification[30] de Firebase permet aux utilisateurs de créer leurs propres comptes via une interface web. Ce service gère l'inscription sécurisée des utilisateurs en collectant leurs informations personnelles telles que nom, prénom, date de naissance, etc. Une fois inscrits, les utilisateurs peuvent se connecter à tout moment pour accéder à leur tableau de bord personnel. L'authentification sécurisée est également utilisée pour limiter l'accès aux informations sensibles telles que les notifications et les alertes.

3.9.3 Présentation des pages de l'interface Web

L'interface utilisateur de notre plateforme web est développée avec HTML5, CSS et JavaScript. Ces technologies offrent une structure flexible pour créer des pages web interactives. Actuellement, plusieurs pages ont été implémentées, mais il est important de noter que ces dernières ne représentent pas une version finale. Ainsi, des modifications ou des ajouts de fonctionnalités restent possibles pour améliorer l'expérience utilisateur ou intégrer de nouvelles exigences selon les besoins.

3.9.3.1 Page d'accueil 3.24

La page d'accueil de l'interface web offre une vue d'ensemble du système, mettant en avant les dernières actualités et annonces importantes. Elle sert également de point d'entrée pour la navigation vers d'autres sections comme l'inscription, la connexion et le tableau de bord.

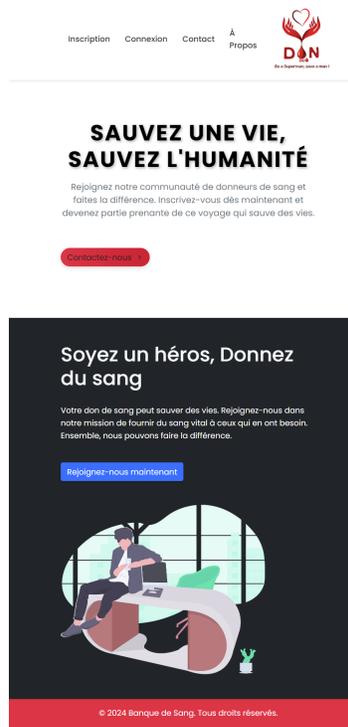


FIGURE 3.24 – Page d'accueil.

3.9.3.2 Page d'inscription 3.25

Sur cette page, les donneurs peuvent créer leur compte en remplissant un formulaire sécurisé. Les données personnelles collectées sont stockées dans Firestore, permettant une gestion centralisée des profils et des informations.

[Accueil](#) [Connexion](#) [Contact](#) [À Propos](#) 

INSCRIVEZ-VOUS

+ [Voir les avantages du compte](#)

Les Avantages du Don de Sang

-  Sauvez des vies en faisant des dons de sang réguliers.
-  Recevez des alertes et notifications en cas d'urgence.
-  Rejoignez une communauté de donneurs engagés.
-  Notifications instantanées pour des besoins de sang urgents.
-  Planification et gestion faciles de vos dons.
-  Active certaines zones du cerveau.
-  Régule le taux de fer dans le corps.
-  Aide à prévenir les maladies cardiovasculaires.

Notre plateforme flexible et simple permet aux donneurs de sauver des vies immédiatement grâce à des alertes et des notifications en temps réel.

Prénom:

Nom:

Date de Naissance: 

Groupe Sanguin:

Wilaya:

Commune:

Maladies (facultatif):

Numéro de Téléphone:

Email:

Mot de Passe:

[S'inscrire](#)

Vous avez déjà un compte ? [Connectez-vous](#)

© 2024 Banque de Sang. Tous droits réservés.

FIGURE 3.25 – Page inscription.

3.9.3.3 Page de connexion 3.26

Les utilisateurs inscrits peuvent se connecter en toute sécurité via cette page en utilisant leurs identifiants (e-mail et mot de passe). L'authentification est gérée par Firebase, garantissant un accès sécurisé et vérifié aux informations personnelles.

Accueil Inscription Contact À Propos



Connectez-vous

Gérer facilement vos rendez-vous,
consulter votre groupe sanguin, accéder
aux résultats de vos examens et suivre
l'historique de vos dons.

Se Connecter

Vous n'avez pas de compte ? [Inscrivez-vous](#)

© 2024 Banque de Sang. Tous droits réservés.

FIGURE 3.26 – Page de connexion.

3.9.3.4 Tableau de bord 3.27

Le tableau de bord personnel des donneurs affiche leurs informations telles que le statut des derniers dons, les rendez-vous planifiés, les notifications et les alertes critiques envoyées via Twilio. Les données sont récupérées en temps réel depuis Firestore, assurant une expérience utilisateur fluide et actualisée.

Sauvez une vie, sauvez l'humanité



Profil

Prénom : wassim	Nom de famille : wali
Date de naissance : 2001-01-01	Groupe sanguin : A-
Sexe : homme	Wilaya : 17
Commune : Djelfa	Maladies : Diabetic
Téléphone : 123456789	Email : main.nail@ail.com

Notifications et Alertes

Date	Heure	Sujet	Contexte
2024-06-21	11:45	Rappel de collecte de sang	Rejoignez-nous au centre communautaire pour notre collecte de sang.
2024-06-21	10:01	Bienvenue !	Vous êtes inscrit avec succès. Bienvenue dans notre communauté de donneurs de sang !

Historique des Dons

Date	Lieu	Quantité (ml)
2024-06-22	Hôpital local	450

Demande de Rendez-vous

Date:

Lieu:

Informations Additionnelles:

[Envoyer la Demande](#)

[Déconnexion](#)

© 2024 Banque de Sang. Tous droits réservés.

FIGURE 3.27 – Tableau de board des donneurs.

3.9.3.5 Page de contact 3.28

Permet aux utilisateurs d'envoyer des commentaires, des questions ou des demandes directement au personnel responsable.

[Accueil](#) [Inscription](#) [Connexion](#) [À Propos](#)



DON
DU
Be a Superman, save a man!

Contactez-nous

Nous sommes là pour vous aider. Remplissez le formulaire ci-dessous et nous vous répondrons dès que possible.

Nom:

Email:

Téléphone:

Message:

📍 Bejaia, Algerie • 📞 +213 1 23 45 67 89 • ✉️
contact@banquedesang.fr
© 2024 Banque de Sang. Tous droits réservés.

FIGURE 3.28 – Page de contact.

3.9.4 Gestion des utilisateurs par le manager

Le manager (responsable médical) accède à une interface spécifique pour gérer tous les utilisateurs (donneurs) inscrits. À travers Firebase, le manager peut visualiser et modifier les profils des donneurs, gérer les notifications et recevoir des alertes en temps réel. Cette fonctionnalité permet une supervision efficace du système de gestion automatisée du sang.

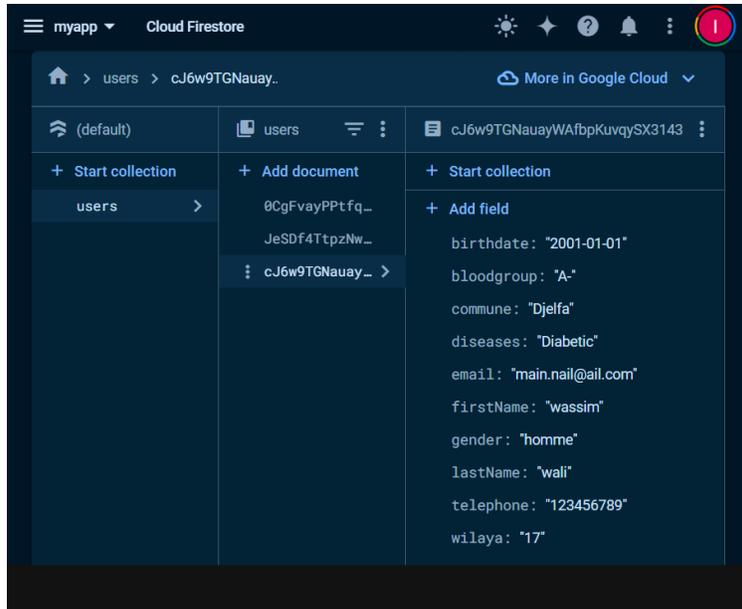


FIGURE 3.29 – Données des donneurs sur Firestore

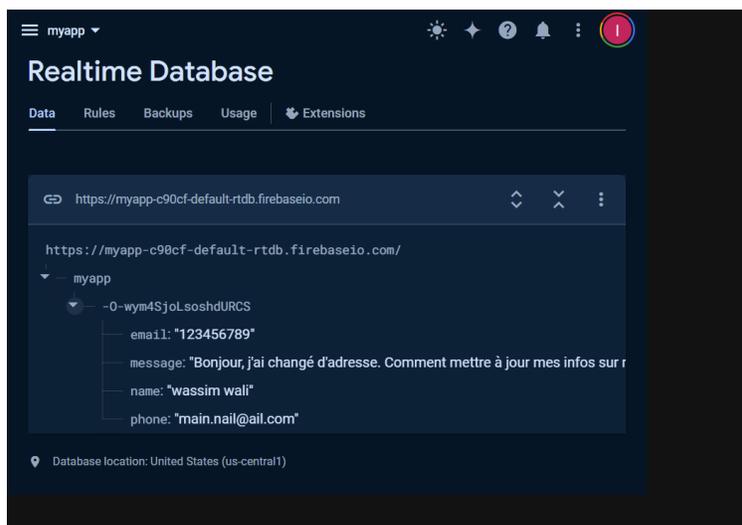


FIGURE 3.30 – Commentaires envoyés par les donneurs via la page de contact.

Remarque : Dans l'interface web précédente, les donneurs recevront des notifications et des alertes de la part des gestionnaires. Cependant, nous n'avons pas pu simuler l'envoi de notifications et de SMS. Malgré cela, nous garantissons que cette plateforme peut fonctionner de manière optimale et qu'elle est conçue pour accepter diverses améliorations futures.

Pour résumer efficacement notre travail présenté dans le chapitre 03, nous avons élaboré le tableau ci-dessous, qui récapitule les principales informations de l'architecture proposée :

TABLEAU 3.1 – Tableau récapitulatif des principaux critères de l'architecture proposée.

Critère	Plateforme IoT pour la gestion automatisée du stockage de sang3
Objectif	Améliorer l'efficacité et la sécurité de la gestion et du stockage automatisé du sang
Population Cible	Centres du stockage de sang, donateurs de sang
Capteurs Utilisés	Capteur de pression Honeywell 26PC, capteur d'humidité et de température SI7021, capteur de tension électrique ZMPT101B
Réseau Utilisé	LoRa, Wi-Fi, MQTT, HTTPS
Techniques de Sécurité	Authentification Firebase
Base de Données	Firebase (Realtime Database, Firestore)
Plateforme	Interface web (HTML5, CSS, JavaScript), Firebase, Twilio pour les notifications SMS
DSS (Decision Support System)	Algorithmes d'analyse des données pour surveiller les niveaux de sang, les conditions environnementales et déclencher des alertes en temps réel
Méthodologie	Surveillance continue des niveaux de sang et des conditions environnementales, envoi de notifications et d'alertes, stockage et analyse des données via le cloud
Résultats Observés	Amélioration de la gestion des stocks de sang, notifications et alertes en temps réel, optimisation des conditions de stockage
Déploiement	Non spécifié
Niveau	Hopitaux, banques de sang

3.10 Conclusion

En conclusion, l'intégration de l'Internet des Objets (IoT) et une plateforme web utilisant Firebase, dans la gestion des banques de sang, représente une avancée remarquable pour répondre aux défis critiques auxquels font face les établissements de santé. L'architecture proposée, combinant capteurs avancés, infrastructure cloud et fonctionnalités web intégrées, permet une surveillance en temps réel des conditions du stockage de sang, garantissant ainsi la disponibilité, la sécurité et la qualité des réserves.

Cela permet non seulement d'assurer une disponibilité optimale du sang nécessaire aux traitements d'urgence, mais aussi facilite la communication entre donateurs et gestionnaires tout en assurant la confidentialité et la fiabilité des données.

3.11 Travaux futurs

Notre architecture de gestion du stockage de sang offre un potentiel considérable pour des améliorations futures, notamment par l'intégration de technologies de machine learning et d'intelligence artificielle (IA). En mettant en œuvre des analyses prédictives, nous pourrions prévoir la quantité de sang nécessaire avant que des pénuries ne se produisent, en analysant les stocks, les demandes de sang et l'arrivée de nouvelles réserves à stocker. Cette approche permettra d'optimiser le stockage et de garantir que les besoins en sang sont satisfaits de manière continue et efficace. De plus, l'IA peut être employée pour la détection d'anomalies afin d'identifier les irrégularités dans les données des capteurs, telles que des changements soudains de température ou d'humidité, indiquant des défaillances potentielles de l'équipement ou d'autres problèmes. Cette capacité permettra des alertes automatiques et une maintenance préventive rapide. En outre, les modèles basés sur l'IA peuvent améliorer le processus de jumelage des donneurs en tenant compte de multiples facteurs tels que le groupe sanguin, la disponibilité des donneurs et l'urgence des receveurs.

L'amélioration de l'expérience utilisateur est également une priorité pour les développements futurs. Le développement d'une application mobile permettra aux donneurs et aux gestionnaires d'accéder à la plateforme en déplacement, facilitant les mises à jour en temps réel, les notifications et la gestion plus facile des profils des donneurs et des rendez-vous. Les améliorations continues de l'interface utilisateur de la plateforme web, basées sur les retours des utilisateurs, la rendront plus intuitive et conviviale, améliorant ainsi l'utilisabilité et la satisfaction globale.

Enfin, l'avancement des technologies de capteurs améliorera encore les capacités du système. La recherche et l'intégration de capteurs plus avancés offrant une plus grande précision et fiabilité pour la surveillance des conditions du stockage de sang garantiront une collecte de données précise et cohérente. L'implémentation de capteurs multi-paramètres capables de mesurer plusieurs conditions simultanément réduira le nombre d'appareils nécessaires et simplifiera la configuration du système, le rendant plus efficace et plus facile à gérer.

Conclusion et perspectives

La présente étude a cherché à explorer la technologie Internet des Objets (IoT) en se concentrant sur son exploitation et son utilisation dans différents domaines. Tout au long de notre analyse, nous avons mis en lumière plusieurs éléments clés qui contribuent à une meilleure compréhension de cette technologie. Ce mémoire a d'abord examiné les principes essentiels de la technologie IoT, puis s'est focalisé sur ses applications spécifiques dans le domaine médical. L'objectif principal était de concevoir une plateforme IoT dédiée à la collecte de données médicales.

La problématique initiale visait à identifier et comprendre les défis et les lacunes persistantes dans le secteur médical, malgré l'abondance des technologies disponibles aujourd'hui. Des lacunes persistantes dans le stockage, la distribution et l'utilisation des produits sanguins exposent les patients à des risques et compromettent la qualité des soins. C'est face à cette situation préoccupante que notre recherche s'est lancée dans une quête de solutions innovantes. Notre objectif premier était d'identifier les défis spécifiques concernant le stockage et la gestion des ressources sanguines, en mettant particulièrement l'accent sur les enjeux critiques liés à leur automatisation.

Nos résultats ont mis en évidence que malgré les avancées technologiques actuelles, plusieurs domaines, notamment le domaine médical, nécessitent encore des développements et améliorations notables. Dans ce contexte, la technologie IoT apparaît comme une solution optimale pour combler ces déficits et répondre à ces besoins croissants.

Dans le premier chapitre, nous établissons les fondements conceptuels et techniques de l'IoT, fournissant une base solide pour comprendre cet univers en pleine expansion. Le second chapitre examine les diverses applications de l'IoT dans différents domaines, notamment celui de la santé, en mettant en avant son impact révolutionnaire sur les soins médicaux. Pour approfondir notre compréhension, nous analysons huit articles de recherche clés, qui illustrent les réalisations et les défis de l'IoT dans ce secteur. Enfin, le troisième chapitre met à profit ces connaissances pour proposer une architecture IoT innovante basée sur le Fog/Cloud computing, optimisant ainsi la gestion automatisée des stocks de sang afin d'assurer à la fois la qualité et la quantité de ce dernier.

La couche Fog assure la collecte des données critiques via des capteurs SI7021 (température et humidité), des capteurs de niveau (quantité de sang) et un capteur ZMPT101B (surveillance de l'alimentation électrique). Un microcontrôleur centralise ces données et les transmet au système Fog computing. Ce dernier exécute un algorithme d'analyse pour vérifier les données par rapport à des seuils critiques prédéfinis. En cas de dépassement de seuil, des actions correctives sont déclenchées (activation du système UPS, allumage d'une alerte rouge). La couche Fog envoie également des notifications (alerte, normal, urgent) au manager via Firebase ou Twilio, en fonction de la gravité de la situation. La couche Cloud, quant à elle, stocke les données des capteurs et les notifications envoyées. Une interface web connectée à Firebase permet aux donneurs de sang de s'inscrire et

de recevoir des notifications de demandes de sang (normales ou urgentes). Un programme de fidélisation est proposé pour encourager les dons de sang (carte). La deuxième partie du mémoire présente une implémentation pratique et une simulation de l'architecture proposée. Elle génère différents types de notifications, présente les résultats d'exécution sous forme de graphiques et valide le bon fonctionnement de l'architecture.

Dans la continuité de nos travaux de mémoire de Master, nous proposons les perspectives suivantes :

- ▷ Mise en place de protocoles de sécurité avancés pour protéger les données sensibles des patients ;
- ▷ Intégrer des algorithmes de machine learning pour prédire les tendances de consommation de sang et optimiser les niveaux de stockage ;
- ▷ Intégrer des systèmes de gestion de l'énergie pour réduire la consommation et prolonger la durée de vie des équipements ;
- ▷ Envisager l'utilisation de sources d'énergie renouvelables comme des panneaux solaires pour alimenter le système UPS ;
- ▷ Implémenter un système d'identification des stocks, comme des codes-barres ou des puces RFID, pour distinguer les nouveaux stocks des anciens ;
- ▷ Proposer une méthode pour traiter les cas de défaillance d'un capteur.

Bibliographie

- [1] https://www.pngegg.com/en/png-crxvk#google_vignette. Consulté le : 24/06/2024 à 18 :40.
- [2] <https://www.oracle.com/internet-of-things/what-is-iot/>. Consulté le : 24/06/2024 à 13 :40.
- [3] <https://cyara.com/blog/pillars-of-iot-user-experience/>. Consulté le : 20/06/2024 à 12 :47.
- [4] <https://kamonk.in/blogs/z-wave-vs-zigbee-your-one-stop-guide/1>. Consulté le : 24/06/2024 à 12 :43.
- [5] <https://www.checkpoint.com/>. Consulté le : 06/06/2024 à 10 :21.
- [6] <https://www.iotforall.com/what-is-a-gateway/>. Consulté le : 06/06/2024 à 10 :25.
- [7] <https://beetroot.co/healthcare/what-is-internet-of-things-in-healthcare>. Consulté le : 05/05/2024 à 9 :07.
- [8] <https://www.wipro.com/business-process/what-can-iot-do-for-healthcare-/>. Consulté le : 05/05/2024 à 10 :34.
- [9] <https://www.tomorrow.bio/fr/poste/comment-l'iot-améliore-le-suivi-des-patients-2023-06-47>. Consulté le : 2024-06-16 à 9 :30.
- [10] https://fr.wikipedia.org/wiki/Donn%C3%A9es_de_sant%C3%A9. Consulté le : 05/05/2024 à 8 :45.
- [11] <https://www.coursera.org/articles/internet-of-things>. Consulté le : 02/04/2024 à 12 :43.
- [12] <https://www.silicon.fr/hub/hpe-intel-hub/les-differentes-couches-dune-infrastructure-iot>. Consulté le : 2024-06-24 à 13 :16.
- [13] <https://www.websiteexample.com/>. Consulté le : 03/04/2024 à 12 :15.
- [14] <https://www.requea.com/iot-gateway.html>. Consulté le : 03/04/2024 à 19 :27.
- [15] <http://www.honeywell.com>. Consulté le : 06/06/2024 à 17 :31.
- [16] <https://www.thelancet.com/journals/langas/home>. Consulté le : 2024-06-16 à 16 :24.
- [17] <https://www.blood.ca/fr>. Consulté le : 2024-06-15 à 22 :48.
- [18] https://fr.wiktionary.org/wiki/Wiktionnaire:Page_d%E2%80%99accueil. Consulté le : 2024-06-15 à 22 :53.

- [19] <https://www.laboandco.com/>. Consulté le : 2024-05-11 à 13 :15.
- [20] <https://www.electroniques.biz/>. Consulté le : 2024-06-10 à 18 :34.
- [21] <https://www.espressif.com/en>. Consulté le : 2024-06-10 à 19 :15.
- [22] <https://www.raspberrypi.org/>. Consulté le : 2024-06-16 à 20 :30.
- [23] <https://www.amazon.com/ELEMENT-Element14-Raspberry-Pi-Motherboard/dp/B07BDR5PDW>. Consulté le : 2024-06-16 à 21 :10.
- [24] <https://www.thethingsnetwork.org/>. Consulté le : 2024-06-09 à 22 :28.
- [25] <https://www.semtech.com/>. Consulté le : 2024-06-06 à 19 :05.
- [26] <https://cloud.google.com/>. Consulté le : 2024-06-09 à 22 :35.
- [27] <https://journalofcloudcomputing.springeropen.com/>. Consulté le : 2024-06-16 à 10 :20.
- [28] <https://www.springeropen.com>. Consulté le : 2024-06-09 à 22 :38.
- [29] <https://techcrunch.com/tag/twilio/1>. Consulté le : 2024-06-16 à 11 :14.
- [30] <https://firebase.google.com/>. Consulté le : 2024-06-15 22 :39.
- [31] <https://www.espressif.com>. Consulté le : 2024-06-14 à 11 :30.
- [32] <https://www.nist.gov/system/files/documents/2017/05/09/bioEvHandbook-poster-rev-06.pdf#:~:text=maintained%20thermostatically%20between%2015.5%C2%B0,with%20less%20than%2060%25%20humidity.&text=equal%20to%20the%20ambient%20temperature,temperature%20and%20humidity%20control%20methods.&text=Sources%3A%20Liquid%20Blood%E2%80%94Farkas%20et%20al>. Consulté le : 2024-06-14 à 11 :30.
- [33] <https://www.jdoodle.com/>. Consulté le : 14/06/2024 à 12 :16.
- [34] <https://fr.rs-online.com/web/>. Consulté le : 14/06/2024 à 12 :40.
- [35] <https://be.farnell.com/fr-BE/silicon-labs/si7021-a20-im1/capteur-humidite-temperature-num/dp/2473673>. Consulté le : 14/06/2024 à 12 :40.
- [36] <https://github.com/Abdurraziq>. Consulté le : 20/06/2024 à 12 :40.
- [37] <https://www.warmanobrien.com/resources/blog/the-top-life-science-companies-utilising-biomet>
Consulté le : 2024-06-23 à 09 :15.
- [38] <https://fr.digi.com/blog/post/iot-in-manufacturing>. Consulté le : 25/06/2024 à 10 :50.
- [39] <https://www.eseye.com/resources/blogs/iot-in-environmental-monitoring-a-breath-of-fresh-air>
Consulté le : 2024-06-24 à 11 :32.
- [40] <https://www.bajajfinserv.in/fitness-tracker-vs-smartwatch>. Consulté le : 2024-06-24 à 12 :01.
- [41] <https://zoe.com/learn/what-is-a-continuous-glucose-monitor>. Consulté le : 2024-06-24 à 10 :00.
- [42] <https://www.abbott.com/corpnewsroom/diabetes-care/revolutionizing-cgm-with-freestyle-libre.html>. Consulté le : 2024-06-24 à 10 :10.

- [43] <https://www.imperial.ac.uk/news/239891/wearable-sensors-styled-into-t-shirts-face/>. Consulté le : 2024-06-24 à 10 :23.
- [44] <https://www.healthrecoveryolutions.com/blog/7-common-remote-patient-monitoring-devices>. Consulté le : 2024-06-24 à 10 :45.
- [45] <https://www.healthrecoveryolutions.com/blog/the-key-benefits-of-telehealth-in-rural-areas>. Consulté le : 2024-06-24 à 10 :49.
- [46] <https://mqtt.org/>. Consulté le : 2024-06-22 à 13 :03.
- [47] <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>. Consulté le : 2024-06-22 à 13 :12.
- [48] <https://xmpp.org/>. Consulté le : 2024-06-22 à 13 :12.
- [49] <https://forum.huawei.com/enterprise/fr/Le-CoAP/thread/791430408733286400-667481008070602752>. Consulté le : 2024-06-22 à 13 :12.
- [50] <https://www.homecontrols.com/home-automation/home-automation-sensors>. Consulté le : 2024-06-25 à 13 :49.
- [51] <https://www.vivint.com/resources/article/home-automation-guide>. Consulté le : 2024-06-25 à 13 :55.
- [52] <https://www.imd.org/blog/digital-transformation/internet-of-things/>. Consulté le : 2024-06-25 à 13 :57.
- [53] <https://www.imd.org/blog/digital-transformation/internet-of-things/>. Consulté le : 2024-06-25 à 14 :05.
- [54] <https://www.productiveedge.com/blog/the-impact-of-internet-of-things-iot-on-healthcare-del>. Consulté le : 2024-06-25 à 14 :11.
- [55] <https://sepiocyber.com/blog/iot-healthcare/>. Consulté le : 2024-06-25 à 13 :47.
- [56] https://www.4medica.com/blog_insights/ensuring-data-quality-in-healthcare-through-effective. Consulté le : 2024-06-25 à 14 :42.
- [57] <https://https://compliance-group.com/the-cia-triad-confidentiality-integrity-availability-f>. Consulté le : 2024-06-25 à 14 :47.
- [58] https://fr.wikipedia.org/wiki/ISO_8601/. Consulté le : 25/06/2024 à 21 :21.
- [59] <https://en.wikipedia.org/wiki/Thermistor/>. Consulté le : 25/06/2024 à 21 :56.
- [60] https://en.wikipedia.org/wiki/Apple_Watch/. Consulté le : 25/06/2024 à 21 :58.
- [61] <https://www.amazon.com/Polar-Bluetooth-Sensor-Fitness-Tracker/dp/B00NOHWT06/>. Consulté le : 25/06/2024 à 22 :01.
- [62] <https://www.amazon.com/Omron-BP786-Pressure-Monitor-Bluetooth/dp/B00T83RDG0/>. Consulté le : 25/06/2024 à 22 :05.
- [63] <https://www.dexcom.com/g6-cgm-system/>. Consulté le : 25/06/2024 à 22 :07.
- [64] <https://www.masimo.com/en-us/>. Consulté le : 25/06/2024 à 22 :09.
- [65] <https://www.resmed.com/en-us/sleep-apnea/devices/airsense-10/>. Consulté le : 25/06/2024 à 22 :12.

- [66] <https://www.fitbit.com/global/us/products/>. Consulté le : 25/06/2024 à 22 :15.
- [67] <https://www.medtronic.com/us-en/healthcare-professionals/products/diabetes/continuous-glucose-monitoring-systems/guardian-sensor-3.html>. Consulté le : 25/06/2024 à 22 :20.
- [68] [https://www.usa.philips.com/healthcare/product/HCNOCTN447/dreamstation-cpap-bi-level-therapy-systems#:~:text=The%20Philips%20Respironics%20DreamStation%20systems,30%20kg%20\(66%20lbs\)](https://www.usa.philips.com/healthcare/product/HCNOCTN447/dreamstation-cpap-bi-level-therapy-systems#:~:text=The%20Philips%20Respironics%20DreamStation%20systems,30%20kg%20(66%20lbs)). Consulté le : 25/06/2024 à 22 :26.
- [69] <https://www.withings.com/us/en/thermo>. Consulté le : 25/06/2024 à 22 :29.
- [70] <https://kardia.com/kardiamobile>. Consulté le : 25/06/2024 à 22 :37.
- [71] <https://www.gehealthcare.com/products/patient-monitoring/patient-monitors/>. Consulté le : 25/06/2024 à 22 :41.
- [72] <https://www.freestylelibre.us/>. Consulté le : 25/06/2024 à 22 :46.
- [73] <https://ouraring.com/>. Consulté le : 25/06/2024 à 22 :48.
- [74] <https://www.withings.com/us/en/scanwatch/>. Consulté le : 25/06/2024 à 22 :53.
- [75] <https://www.dexcom.com/en-us/g7-cgm-system>. Consulté le : 25/06/2024 à 22 :56.
- [76] <https://www.philips.ca/healthcare/product/HC989803203011/biosensor-bx100-wearable-remote-measurement-device>. Consulté le : 25/06/2024 à 22 :59.
- [77] <https://www.biointellisense.com/>. Consulté le : 25/06/2024 à 23 :26.
- [78] <https://justdoelectronics.com/zmpt101b-voltage-sensor>. Consulté le : 25/06/2024 à 8 :01.
- [79] https://colab.research.google.com/drive/12tC-XsZr10NAxUPLncA_XMX7txF0aV7b?usp=sharing. Consulté le : 26/06/2024 à 15 :59.
- [80] <https://colab.research.google.com/>. Consulté le : 26/06/2024 à 18 :59.
- [81] Philippe Atelin. *Wi-Fi : réseaux sans fil 802.11*. Ediciones ENI, 2008.
- [82] Leire Bastida, Gloria Cea, Ana Moya, Alba Gallego, Eugenio Gaeta, Sara Sillaurren, Paulo Barbosa, Sabrina Souto, Eujessika Rodrigues, Macarena Torrego-Ellacuría, et al. Promoting obesity prevention and healthy habits in childhood : the ocariot experience. *IEEE Journal of Translational Engineering in Health and Medicine*, 11 :261–270, 2023.
- [83] William Bolton. *Instrumentation and Control Systems*. 2004.
- [84] Aymeric Bouchereau. Appréhender la réalité de l'internet des objets dans le cadre de l'apprentissage humain. In *SemDoc du réseau IRIS 2018*, 2018.
- [85] Alexandros-Apostolos A Boulogeorgos, Panagiotis D Diamantoulakis, and George K Karagiannidis. Low power wide area networks (lpwans) for internet of things (iot) applications : Research challenges and future trends. *arXiv preprint arXiv :1611.07449*, 2016.
- [86] M. Chen, Y. Li, and C. D. Nugent. Internet of things (iot) in healthcare : A comprehensive review. *Journal of Medical Systems*, 42(12) :1–17, 2018.

- [87] Danyllo V da Silva, Taisa G Gonçalves, and Paulo F Pires. Using iot technologies to develop a low-cost smart medicine box. In *Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 97–101. SBC, 2019.
- [88] Mehdi Dadkhah, Mohammad Mehraeen, Fariborz Rahimnia, and Khalil Kimiafar. Use of internet of things for chronic disease management : an overview. *Journal of Medical Signals & Sensors*, 11(2) :138–157, 2021.
- [89] Robertas Damaševičius, Nebojsa Bacanin, and Sanjay Misra. From sensors to safety : Internet of emergency services (ioes) for emergency response and disaster management. *Journal of Sensor and Actuator Networks*, 12(3) :41, 2023.
- [90] Warren Gay. *Mastering the raspberry PI*. Apress, 2014.
- [91] Drew Gislason. *Zigbee Wireless Networking*. 2008.
- [92] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy : An emerging low-power wireless technology. *sensors*, 12(9) :11734–11753, 2012.
- [93] Renaldi Gondosubroto. *Internet of Things from Scratch : Build IoT Solutions for Industry 4.0 with ESP32, Raspberry Pi, and AWS*. Packt Publishing, 1st edition edition, 2024.
- [94] Bogdan Stefanescu Guillaume Villemaud. Smart anything - comment alimenter des milliards d'objets connectés?, 2020.
- [95] Mohamed Tahar Hammi. *Sécurisation de l'Internet des objets*. PhD thesis, Université Paris Saclay (COMUE), 2018.
- [96] Darko Hercog, Tone Lerher, Mitja Truntič, and Oto Težak. Design and implementation of esp32-based iot devices. *Sensors*, 23(15) :6739, 2023.
- [97] Evelyn J. S. Hovenga and Heather Grain. Health data and data governance. *Studies in Health Technology and Informatics*, 192 :826–830, 2013.
- [98] Kashif Ishaq and Syed Shah Farooq. Exploring iot in smart cities : Practices, challenges and way forward. *arXiv preprint arXiv :2309.12344*, 2023.
- [99] Md Reazul Islam, Md Mohsin Kabir, Muhammad Firoz Mridha, Sultan Alfarhood, Mejdil Safran, and Dunren Che. Deep learning-based iot system for remote monitoring and early detection of health issues in real-time. *Sensors*, 23(11) :5204, 2023.
- [100] Anthony JUTON. Réseaux très basse consommation, longue portée, bas débit, l'exemple de lorawan. *Revue 3EI*, 2019.
- [101] Dae-Young Kim, Young-Sik Jeong, and Seokhoon Kim. Data-filtering system to avoid total data distortion in iot networking. *Symmetry*, 9(1) :16, 2017.
- [102] Agus Kurniawan. *LoRa and LoRaWAN for IoT : Enabling Low-Power Wide-Area Networks*. 2020.
- [103] Chia-Yen Lee and Gwo-Bin Lee. Humidity sensors : a review. *Sensor Letters*, 3(1-2) :1–15, 2005.
- [104] Pedro Maia, Thais Batista, Everton Cavalcante, Augusto Baffa, Flavia C Delicato, Paulo F Pires, and Albert Zomaya. A web platform for interconnecting body sensors and improving health care. *Procedia Computer Science*, 40 :135–142, 2014.
- [105] Serigne Modou Kara Mbengue. Conception et implémentation d'une application d'iomt pour le diagnostic et le suivi du diabète. 2020.

- [106] Larras Melissa and Khalfouni Djamila. *Défis de sécurité de l'Internet des Objets Problèmes et solutions*. PhD thesis, Université Mouloud Mammeri, 2019.
- [107] Huda Hussein Mohamad Jawad, Zainuddin Bin Hassan, Bilal Bahaa Zaidan, Farah Hussein Mohammed Jawad, Duha Husein Mohamed Jawad, and Wajdi Hamza Dawod Alredany. A systematic literature review of enabling iot in healthcare : Motivations, challenges, and recommendations. *Electronics*, 11(19) :3223, 2022.
- [108] Bhabendu Kumar Mohanta, Debasish Jena, Utkalika Satapathy, and Srikanta Patnaik. Survey on iot security : Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet of Things*, 11 :100227, 2020.
- [109] Mohamed Khaled Mohyeldin Naeim, Gwo Chin Chung, It Ee Lee, Jun Jiat Tiang, and Soo Fun Tan. A mobile iot-based elderly monitoring system for senior safety. *International Journal of Technology*, 14(6) :1185, 2023.
- [110] Ambika Nagaraj. *Introduction to sensors in IoT and cloud computing applications*. Bentham Science Publishers, 2021.
- [111] Firas Ouerghi. Internet of things for ehealth. Technical report, 2018.
- [112] Octavian Adrian Postolache, Edward Sazonov, and Subhas Chandra Mukhopadhyay. *Sensors in the Age of the Internet of Things : Technologies and Applications*. The Institution of Engineering and Technology (IET), 2019.
- [113] Dalia Kamal Abd Alla Rizk, Hoda M Hosny, Sayed ElHorbety, and Abdel-Badeeh Salem. Smart hospital management systems based on internet of things : challenges, intelligent solutions and functional requirements. *International Journal of Intelligent Computing and Information Sciences*, 22(1) :32–43, 2022.
- [114] S. Selvaraj and S. Sundaravaradhan. Challenges and opportunities in iot healthcare systems : a systematic review. *SN Applied Sciences*, 2(1) :139, 2020.
- [115] Amrita Shahani, Harry Reyes Nieva, Katie Czado, Evan Shannon, Raquel Gaetani, Marcus Gresham, Jose Cruz Garcia, Hareesh Ganesan, Emily Cerciello, Janan Dave, et al. An electronic pillbox intervention designed to improve medication safety during care transitions : challenges and lessons learned regarding implementation and evaluation. *BMC health services research*, 22(1) :1304, 2022.
- [116] Hooman Sharifnasab, Ali Mahrokh, Hossein Dehghanisani, Ewa Łazuka, Grzegorz Łagód, and Hamed Karami. Evaluating the use of intelligent irrigation systems based on the iot in grain corn irrigation. *Water*, 15(7) :1394, 2023.
- [117] Hitesh Kumar Sharma, JC Patni, Prashant Ahlawat, and Siddhratha Sankar Biswas. Sensors based smart healthcare framework using internet of things (iot). *Int J Sci Technol Res*, 9(2) :1228–34, 2020.
- [118] Paolo Bonato Leighton Chan Shyamal Patel, Hyung Park and Mary Rodgers. A review of wearable sensors and systems with application in rehabilitation. *IEEE Transactions on Biomedical Engineering*, 59(9) :2201–2220, 2012.
- [119] Ian R. Sinclair and John Dunton. *Sensors and Transducers*. 2001.
- [120] Christos L Stergiou, Andreas P Plageras, Vasileios A Memos, Maria P Koidou, and Konstantinos E Psannis. Secure monitoring system for iot healthcare data in the cloud. *Applied Sciences*, 14(1) :120, 2023.

- [121] Saumendra Pattnaik Suprava Ranjan Laha, Binod Kumar Pattanayak. Advancement of environmental monitoring system using iot and sensor : A comprehensive analysis. *International Journal of Advanced Computer Science and Applications*, 12(5) :123–131, 2021.
- [122] Navod Neraanjan Thilakarathne, Mohan Krishna Kagita, and Thippa Reddy Gadekallu. The role of the internet of things in health care : a systematic and comprehensive study. *Available at SSRN 3690815*, 2020.
- [123] Ankush Singh Tomar, Shashank Patil, Sandeep Sharma, and Shantanu Chakraborty. Behavioral data and lifestyle changes : Impact on health outcomes. *National Center for Biotechnology Information*, 2021.
- [124] Nouhoum Ibrahim Traoré, Mariam Tiotio Berthe, and Abdelhak Aqqal. Vers la mise en place d’une plateforme iot d’aide au diagnostic des maladies neuromusculaires via les smartphones comme objets connectés. In *Colloque sur les Objets et systèmes Connectés*, 2019.
- [125] Banu Çaliş Uslu, Ertuğ Okay, and Erkan Dursun. Analysis of factors affecting iot-based smart hospital design. *Journal of Cloud Computing*, 9(1) :67, 2020.
- [126] Zhibin Wang, Xiaojun Qiao, Ying Wang, Hao Yu, and Cuixia Mu. Iot-based system of prevention and control for crop diseases and insect pests. 15, 2024.
- [127] John G. Webster. *The Measurement, Instrumentation and Sensors Handbook*. Springer, 1999.
- [128] John G. Webster and Amit J. Nimunkar. *Principles of Measurement and Transduction of Biomedical Variables*. 2015.
- [129] Jon S Wilson. *Sensor technology handbook*, volume 1. Newnes, 2005.
- [130] Dhanalekshmi Prasad Yedurkar, Shilpa P Metkar, Fadi Al-Turjman, Thompson Stephan, Manjur Kolhar, and Chadi Altrjman. A novel approach for multichannel epileptic seizure classification based on internet of things framework using critical spectral verge feature derived from flower pollination algorithm. *Sensors*, 22(23) :9302, 2022.
- [131] Emre Yıldırım, Murtaza Cicioğlu, and Ali Çalhan. Fog-cloud architecture-driven internet of medical things framework for healthcare monitoring. *Medical & Biological Engineering & Computing*, 61(5) :1133–1147, 2023.
- [132] Jinghui Zhang, Xinyang Li, Junhe Li, Qiangsheng Dai, Zhen Ling, and Ming Yang. Bluetooth low energy device identification based on link layer broadcast packet fingerprinting. *Tsinghua Science and Technology*, 28(5) :1–11, 2023.

RÉSUMÉ

Notre projet explore le conception d'une architecture IoT combinée à une plateforme web pour la gestion automatisée du stockage de sang , visant à améliorer l'efficacité et la sécurité de stockage de différents types de sang. Dans la première partie de notre travail, nous avons mis en place un réseau de capteurs reliés à un microcontrôleur qui est connecté à un système cloud et un fog computing. Ce réseau permet de surveiller en continu les conditions de stockage du sang, telles que le volume disponible par groupe sanguin, ainsi que les conditions environnementales comme la température et l'humidité. Les données recueillies sont traitées localement par le fog avant d'être envoyées au cloud pour un traitement et une analyse plus approfondie. La seconde partie se focalise sur le développement d'une plateforme web destinée aux donneurs de sang. Cette plateforme, intégrée avec Firebase, permet aux utilisateurs de gérer leur profil, de programmer des rendez-vous de don, de consulter l'historique de leurs dons, et de recevoir des notifications et alertes en cas de manque de sang . La plateforme utilise Firebase pour stocker les données des donneurs et des capteurs, garantissant ainsi la sécurité et la disponibilité des informations. Enfin, la troisième partie de notre projet a consisté en la simulation de cette architecture dans un environnement contrôlé pour valider le fonctionnement et l'efficacité du système proposé, en assurant une gestion optimale des réserves de sang grâce à l'automatisation et à la supervision continue.

Mots clés : IoT, Fog et Cloud Computing, Plateforme Web, Firebase.

ABSTRACT

project explores the design of an IoT architecture combined with a web platform for the automated management of blood storage, aiming to improve the efficiency and safety of storing different types of blood. In the first part of our work, we implemented a network of sensors connected to a microcontroller that is linked to a cloud system and fog computing. This network allows continuous monitoring of blood storage conditions, such as the available volume per blood group, as well as environmental conditions like temperature and humidity. The collected data is processed locally by the fog before being sent to the cloud for further processing and analysis. The second part focuses on the development of a web platform intended for blood donors. This platform, integrated with Firebase, allows users to manage their profiles, schedule donation appointments, view their donation history, and receive notifications and alerts in case of blood shortages. The platform uses Firebase to store donor and sensor data, ensuring the security and availability of the information. Finally, the third part of our project involved simulating this architecture in a controlled environment to validate the functionality and efficiency of the proposed system, ensuring optimal management of blood reserves through automation and continuous supervision.

Key words : IoT, Fog and Cloud Computing, Web Platform, Firebase.