

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la Matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes informatique

Thème :

**Plateforme de développement pour l'internet des objets (IdO)
avec un apprentissage automatique**

Encadré par :

Dr.Kouahla Zineddine

Présenté par :

Lokba M.Zine Laabidine

Juillet 2019

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

Avant tout je tiens à remercier Dieu tout puissant de m'avoir donné la force et le courage pour surmonter toutes les difficultés rencontrées durant ces longues années d'études.

Il est très agréable de réserver cette page comme un témoignage de reconnaissance à toutes les personnes qui m'ont aidés pour réaliser ce travail.

En préambule à ce mémoire.

Je souhaite adresser mes remerciements les plus sincères à mes parents et aux personnes qui m'ont apporté leur aide, leur assistance et qui ont contribué à l'élaboration de ce mémoire, ainsi qu'à la réussite de cette formidable année universitaire.

Je tiens à exprimer sincèrement mon profonde gratitude et remerciement à

Dr.Kouahla Zineddine & Pr.Aouamer Messaoud,

en tant qu'encadreurs pour leurs soutien amical et constant,

pour m'avoir encadré, guidé et conseillé avec rigueur et enthousiasme tout au long de la réalisation de ce mémoire, ainsi que pour l'inspiration qu'il m'a donné.

Je tiens à remercier aussi Dr.Medjaldi Tayeb,

qui a fait l'entente entre notre université et l'université de Québec(Canada),

où j'ai fait mon stage, ainsi pour ces conseils et remarques précieuses.

Je remercie également toute l'équipe du laboratoire LSSI pour leur accueil et leur esprit d'équipe.

Aux membres de jury qui m'ont fait l'honneur d'évaluer mon travail.

Enfin, je m'adresse mes plus sincères remerciements à

tous mes proches et amis,

qui m'ont toujours soutenues et encouragés au cours de la réalisation de ce mémoire.

Merci à vous tous.

Résumé

La révolution des technologies de l'information et de la communication (TIC) a créé aujourd'hui le concept d'*Internet des Objets* (IdO) qui est considéré comme l'extension du réseau internet où il devient très répandu dans différents domaines. Avec des périphériques à faible consommation d'énergie et de bande passante, la localisation en intérieur devient un sujet intéressant. Les algorithmes d'apprentissage automatique sont largement utilisés pour estimer des localisations. L'un des principaux problèmes des systèmes de positionnement d'intérieur est de trouver l'algorithme d'apprentissage automatique approprié. Dans ce mémoire, nous avons réalisé un prototype d'une plateforme Internet des Objets afin d'examiner de plus près les aspects sous-jacents de la localisation en intérieur basée sur des données de signaux tel que l'indicateur de puissance du signal (RSSI), le rapport signal sur bruit (SNR), la fréquence (FQ), ainsi sur des données environnementales tel que la température dans les réseaux IdO à longue portée tel que LoRaWAN. Des algorithmes d'apprentissage automatique et profond ont été comparés en termes de précision afin d'identifier le meilleurs parmi eux.

Mots clés : Internet des objets, apprentissage automatique, apprentissage profond, localisation

Abstract

The revolution in information and communication technologies (ICT) has today created the concept of the Internet of Things (IoT), which is considered as the extension of the Internet network where it is becoming very widespread in different fields. With devices that consume little energy and bandwidth, indoor location becomes an interesting subject. Machine learning algorithms are widely used to estimate localization. One of the main problems with indoor localization systems is to find the appropriate machine learning algorithm. In this work, we have developed a prototype of an Internet of things platform to examine more closely the underlying aspects of indoor localization based on signal data such as signal strength indicator (RSSI), signal to noise ratio (SNR), frequency (FQ), and environmental data such as temperature in long-range IdO networks such as LoRaWAN. Machine and deep learning algorithms were compared in terms of accuracy to identify the best among them.

Key words: Internet of Things, Machine Learning, Deep Learning, Localization.

Table des matières

Introduction Générale.....	1
----------------------------	---

Chapitre 1 : Internet des Objets & Cloud Computing

1. Introduction.....	3
2. L'Internet des Objets (IdO)	3
2.1 Définitions.....	3
3. L'infrastructure de l'IdO	5
4. Fonctionnement de l'IdO.....	6
5. Les applications de l'IdO	7
5.1 La maison intelligente « Smart Home ».....	8
5.2 Soins de santé.....	9
5.3 Transport	9
5.4 L'énergie	9
5.5 L'environnement.....	9
6. Cloud Computing	10
6.1 Définition	10
6.2 Les types de Cloud Computing	11
6.3 Les modèles du Cloud.....	11
7. Conclusion.....	13

Chapitre 2 : L'apprentissage Automatique

1. Introduction	14
2. L'apprentissage automatique	14
2.1 Définition	14
2.2 Les algorithmes d'apprentissage automatique.....	14
2.2.1 Le voisin K le plus proche (K- nearest neighbours).....	14

2.2.2	Machines à vecteurs de support (SVM)	15
2.2.3	Arbre de décision (Decision Tree)	15
2.2.4	Forêts aléatoire	16
2.2.5	Réseaux de neurones	16
3.	L'utilisation des algorithmes d'apprentissage automatique dans l'IdO	17
4.	Conclusion.....	19

Chapitre 3 : Système embarqué

1.	Introduction	20
2.	Système embarqué.....	20
2.1.	Architecture d'un système embarqué	20
3.	Carte d'interface	21
4.	Exemples des cartes d'interface	22
5.	Famille de la carte Minized	24
6.	Conclusion.....	25

Chapitre 4 : Étude Conceptuelle

1.	Introduction	26
2.	Objectif du projet et méthodologie.....	26
3.	Méthode de conception adoptée	27
4.	Architecture générale du système.....	27
4.1	Interfaces du système	27
4.2	Composants physique	28
4.2.1.	Capteurs.....	28
4.2.2.	Microcontrôleur.....	28
4.2.3.	Passerelle	29
4.2.4.	Carte Transmetteur/Récepteur (Transceiver)	29
4.3	Déroulement et fonctionnement.....	29
5.	Configuration physique de l'endroit où on a fait l'expérimentation.....	30
6.	Principe de reconnaissance du contexte de localisation	31

7. Architecture détaillé des composants du système	32
7.1 Interface physique	32
7.1.1 Interface pour la capture des données brutes du contexte	33
7.1.2 Interface IdO pour la transmission	34
7.1.3 Scénario d'application	34
7.2 Interface d'apprentissage	35
7.2.1 L'apprentissage automatique.....	35
7.2.2 L'apprentissage profond.....	37
8. Conclusion.....	38

Chapitre 5 : Implémentation

1. Introduction	39
2. Implémentation de l'interface physique de capture	39
2.1 Les composants principaux	40
3. Composants du système	41
3.1 Principe de raccordement.....	44
4. Implémentation du Module de contrôle et de commande	44
4.1 Environnement d'exécution.....	44
4.2 Outils et langage de développement	45
4.2.1 Outils de développement.....	45
4.2.2 Langages de développement	46
5. Résultats et discussion.....	46
5.1 Résultats de l'apprentissage automatique	47
5.1.1 Résultats d'application sur les données de signaux.....	47
5.1.2 Résultats d'application sur les données de signaux et environnementaux	49
5.2 Résultats de l'apprentissage profond	50
6. Conclusion.....	51
Conclusion Générale	52
Bibliographie.....	53

Liste des Figures

Figure 1.1 L'infrastructure de l'IdO.....	6
Figure 1.2 Les principaux domaines d'application de l'IdO.).....	8
Figure 1.3 Les principaux modèles du Cloud.....	13
Figure 2.1 La classification avec KNN	15
Figure 2.2 Architecture d'un arbre de décision	16
Figure 2.3 Architecture d'un réseau de neurones artificiel avec une seule couche cachée	17
Figure 2.4 Taxonomie des algorithmes d'apprentissage automatique.....	19
Figure 3.1 Schéma d'un système embarqué typique.....	20
Figure 3.2 La carte Minized	24
Figure 4.1 Architecture générale du système	28
Figure 4.2 Diagramme de séquence du fonctionnement du système	29
Figure 4.3 Diagramme de séquence correspond au scénario détaillé d'automatisation.....	30
Figure 4.4 Plan de l'expérimentation	31
Figure 4.5 Représentation de la relation entre les données et le contexte de localisation.....	32
Figure 4.6 Architecture détaillé du système	32
Figure 4.7 Diagramme d'activité correspond au sous-système capture	33
Figure 4.8 Diagramme d'activité correspond au sous-système de transmission.....	34
Figure 4.9 Déroulement de l'application.....	35
Figure 4.10 Schéma du fonctionnement d'un neurone	38
Figure 5.1 Conception matériel du microcontrôleur	39
Figure 5.2 Le contrôleur AXI BRAM	40
Figure 5.3 Le Bus I2C	40
Figure 5.4 Le port GPIO.....	41

Figure 5.5 Le capteur HTU21D	41
Figure 5.6 Le microcontrôleur Minized	42
Figure 5.7 La carte Tx/Rx LL-RLP-20	42
Figure 5.8 L'interface de cloud conductor.....	43
Figure 5.9 Le Passerelle LinkLabs	43
Figure 5.10 Schéma de raccordement LL-RLP-20 et Minized	44
Figure 5.11 Matrice de confusion de KNN	47
Figure 5.12 Matrice de confusion de SVM	47
Figure 5.13 Matrice de confusion de DT	48
Figure 5.14 Matrice de confusion de RF.....	48
Figure 5.15 Matrice de confusion de KNN	49
Figure 5.16 Matrice de confusion de SVM	49
Figure 5.17 Matrice de confusion de DT	49
Figure 5.18 Matrice de confusion de RF.....	49
Figure 5.19 Résultats de précision et de fonction de perte d'un réseau de neurones	51

Liste des Tableaux

Tableau 2.1 Taxonomie des travaux de recherche sur les algorithmes d'apprentissages automatique	18
Tableau 3.1 Les cartes d'interface les plus connu	23
Tableau 5.1 Caractéristiques du pc utilisé pour le projet	45
Tableau 5.2 Résultats d'application des algorithmes d'apprentissage sur les données.....	48
Tableau 5.3 Résultats d'application des algorithmes d'apprentissage sur les données de signaux et environnementaux	50

Introduction Générale

La révolution des technologies de l'information et de la communication (TIC) a créé aujourd'hui le concept d'*Internet des Objets* (IdO) qui est considéré comme l'extension du réseau Internet où il devient très répandu dans différents domaines. Ce réseau émergent permet à des milliards d'objets physiques ou virtuels (voiture, lunettes, montres, Smart-Phone, etc) de se connecter et de communiquer entre eux via Internet de manière intelligente grâce à des protocoles de communication standard. (Yasmin Fathy et al, 2018).

Avec ce grand nombre d'objets, la quantité de données numériques qui est capturée par ces objets a explosé. Cette expansion a créé le besoin de stocker, gérer et sécuriser d'énormes volumes de données afin de tirer la meilleure partie de cette masse d'informations pour un public de plus en plus vaste. Pour relever ces défis, un nouveau paradigme informatique appelé *Cloud Computing* est nécessaire, dans lequel les données et les services sont disponibles de manière flexible et transparente.

L'Internet des Objets a évolué depuis et est devenu de plus en plus populaire ces dernières années à la convergence de plusieurs technologies, telles que les microcontrôleurs, les capteurs, les systèmes embarqués et les communications sans fil.

La mise en réseau sans fil de l'Internet des Objets avec longue portée, faible consommation d'énergie et faible débit apparaît comme un nouveau paradigme permettant de connecter efficacement des milliers de périphériques. Dans de tels réseaux avec des périphériques à faible consommation d'énergie et de bande passante, la localisation devient un sujet intéressant. (Hazem Sallouha et al, 2017).

L'objectif de ce mémoire, d'une part est de réaliser un prototype d'une plateforme IdO qui comporte un système embarqué dedans, ce dernier permet la transmission des données capturées par cette plateforme à travers un protocole de l'internet des objets vers le cloud.

D'autre part, les algorithmes d'apprentissage automatique font partie de notre solution afin d'avoir une meilleure précision pour la prédiction de la localisation en intérieur, en se basant sur des données de signaux tel que l'indicateur de puissance du signal (RSSI), le rapport signal sur bruit (SNR), la fréquence (FQ), ainsi sur des données environnementales tel que la température et l'humidité dans les réseaux IdO à longue portée tel que LoRaWAN.

Autre que l'introduction et la conclusion générale, notre mémoire contient cinq chapitres :

Chapitre 1 concerne *l'Internet des Objets* et le *Cloud Computing*. Dans ce chapitre, nous définissons *l'Internet des Objets*, nous présentons son infrastructure, son méthode de fonctionnement ainsi nous décrivons ses différents domaines d'application. Concernant le *Cloud Computing*, nous le définissons, et nous présentons leurs types et leurs modèles.

Chapitre 2 se concentre sur *l'apprentissage automatique*. Dans cet état de l'art, nous présentons ce concept, ses différents algorithmes ainsi que son utilisation dans la littérature et plus précisément dans le domaine de *l'Internet des Objets*.

Chapitre 3 est dédiées aux *systèmes embarqués* qui jouent un rôle important dans notre travail. Dans ce chapitre nous définissons les systèmes embarqués, nous décrivons leur architecture et nous présentons la carte d'interfaçage ainsi que celle choisie pour ce travail.

Chapitre 4 est consacré à *l'étude conceptuelle* de notre système en spécifiant la méthode utilisée pour le représenter, définir ces fonctionnalités pertinente, une vue générale du système qui nous permettra de le comprendre, puis on entre un peu dans le vif du sujet on spécifiant les composants de base de notre système par une architecture détaillée et bien appropriée

Chapitre 5 dans ce dernier chapitre nous montrerons la structuration du système embarqué, les outils utilisés pour développer notre système ainsi que les différents résultats atteints.

En dernier lieu, une conclusion générale clôtura ce mémoire et synthétise le travail réalisé et les perspectives envisagés.

Chapitre 1

Internet des Objets & Cloud Computing

1. Introduction

Aujourd'hui, IdO est plus un terme descriptif pour une vision selon laquelle tout doit être connecté à Internet. L'IdO sera essentiel dans le futur car le nouveau concept ouvre des opportunités pour de nouveaux services et innovations. (Evans Dave, 2011), a étudié l'augmentation du nombre d'objets connectés à Internet et a constaté qu'en 2020, il y aurait 50 milliards d'objets connectés, soit 7 objets par personne. Ce grand nombre d'objets produit en permanence une quantité énorme de données qui nécessite une puissance de calcul importante et une capacité de stockage incroyable pour l'analyse et le stockage. Pour satisfaire ces exigences, l'adaptation de *Cloud Computing* pourrait être rentable. (Rottenberg Sam, 2015) a noté que cette adaptation faciliterait l'utilisation de leurs performances de traitement et de leur capacité de stockage les plus élevées (Chunye Gong et al, 2010).

Dans ce chapitre, nous présentons l'état de l'art de l'Internet des Objets ainsi que sur le paradigme du *Cloud Computing*, dans lequel, dans la première partie, nous passons en revue les concepts et les définitions de l'état de la technique liée à l'Internet des objets, ses applications, son infrastructure, et sa méthode de fonctionnement. Ensuite, en ce qui concerne la deuxième partie « Cloud », nous le définissons, et présentons ces types et ces modèles.

2. L'Internet des Objets (IdO)

2.1. Définitions

Jusqu'à nos jours, il est difficile de trouver une définition claire et précise pour l'Internet des Objets (IdO). Nous avons donc rassemblé quelques définitions déjà existantes dans la littérature.

Définition 1

« L'Internet des Objets (IdO) est une partie de future Internet et peut être défini comme une infrastructure de réseau mondial dynamique avec capacité de configuration automatique basée sur des protocoles de communication standard et interopérables ou les objets ont des identités, des attributs physiques et des personnalités virtuelles et utilisent des interfaces intelligentes, et sont intégrés de façon transparente dans le réseau d'information. Ces objets peuvent interagir et communiquer entre eux et avec l'environnement en échangeant des données et des informations sensibles à l'environnement tout en réagissant de manière

autonome aux événements et les influençant en exécutant des processus qui déclenchent des actions et créent des services avec ou sans intervention humaine » (Vermesan Ovidiu et al, 2011)

Définition 2

“ L'idée de base de ce concept est que la présence omniprésente autour de nous d'une variété de choses ou d'objets - comme les étiquettes d'identification par radiofréquence (RFID), capteurs, actionneurs, téléphones portables, etc. - qui, grâce à des schémas d'adressage uniques, sont capables d'interagir les uns avec les autres et de collaborer avec leurs voisins pour atteindre des objectifs communs. “ (Atzori Luigi et al, 2010).

Définition 3

“ Le groupe de travail Internet of Things Global Standards Initiative (IoT-GSI), piloté par l'Union Internationale des Télécommunications (UIT) considère l'IdO comme « une infrastructure mondiale au service de la société de l'information » permettant « d'offrir des services évolués en interconnectant des objets (physiques et virtuels) grâce à l'interopérabilité de technologies de l'information et de la communication existantes ou en évolution “ (Christophe Baland et al, 2016).

Définition 4

“L'Internet des Objets est un réseau des réseaux qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant” (Pierre-Jean Benghozi et al, 2008).

Définition 5

“Le terme IdO est utilisé comme un mot-clé pour couvrir divers aspects liée à l'extension de l'internet et du web dans le domaine physique, grâce au déploiement généralisé de dispositifs spatialement distribue avec identification intégrée, capacités de détection et/ou d'actionnement “ (Daniele Miorandi et al, 2012).

De notre point de vue, nous constatons que "l'Internet des Objets est un réseau d'objets, généralement des capteurs et des actionneurs sans fil, qui sont identifiables de façon unique dans le monde, fonctionnent dans un environnement intelligent. Ces objets peuvent interagir et communiquer entre eux et avec l'environnement par le biais de protocoles de communication standard faisant appel à la technologie de l'information et de la communication pour l'échange de données et d'informations qui sont généralement stockées dans le Cloud".

3. L'infrastructure de L'IdO

Dans notre cadre de travail **l'Infrastructure** est le minimum d'éléments nécessaires à la réalisation d'une application basée sur l'IdO.

- **Les objets objets** inclus dans IdO ne sont pas seulement des appareils complexes tels que les smartphones et les tablettes, mais ils comprennent également des objets du quotidien, ces objets physiques possèdent des technologies embarquées de capteurs, d'intelligence et de connectivité, leur permettant de communiquer avec d'autres objets, qui apportent la connectivité aux objets via des composants matériels et logiciels embarqués. On trouve parmi ces acteurs les électroniciens, les fondeurs de semi-conducteurs, les fabricants de capteurs ou encore les développeurs de logiciels embarqués assurant la connectivité (Arcep, 2016).
- Les types de **réseau** qui viennent immédiatement à l'esprit sont le Bluetooth, le Wifi et le couple 3G/4G. Wifi et Bluetooth ont une portée limitée. Ce qui ne les rend pas inutiles, bien au contraire. Dans un environnement restreint et bien pourvu en sources d'énergie. La 3G/4G est à plus grande portée, mais demande là encore beaucoup d'énergie, mais faciles à mettre en œuvre. Cependant il existe des nouvelles technologies qui sont étendus et qui ont une longue portée [1].
- Les opérateurs de **Cloud** assurent principalement le stockage, analyse et le traitement des données brutes. Sur ce segment, les acteurs historiques de l'Internet et des serveurs sont concurrencés par de nouveaux acteurs déployant leurs propres infrastructures informatiques. (Arcep, 2016)

- Les fournisseurs **d'interfaces** logicielles, ou de middleware, permettant de faire communiquer les différents objets. On retrouve sur ce segment des éditeurs traditionnels de logiciels. (Arcep, 2016).
- Les acteurs de la **sécurité**, présents à tous les niveaux de la chaîne, depuis la conception de l'objet jusqu'aux services. Dans le meilleur des cas, ces acteurs issus du monde de la sécurité informatique travaillent en étroite collaboration avec tous les acteurs de la chaîne de valeur. Certains sont par ailleurs absorbés par les acteurs de l'Internet des Objets. (Arcep, 2016).



Figure 1.1 L'infrastructure de l'IdO.

4. Le fonctionnement de l'IdO

Pour assurer le bon fonctionnement de l'Internet des Objets (IdO), il existe plusieurs étapes à respecter. Perry Xiao (2018) les a résumés dans cinq étapes notamment:

- **Tout d'abord**, chaque « objet » sur l'Internet des Objets doit avoir une identité unique. Grâce à l'évolution de l'adresse IP (Internet Protocol) et ses générations on peut fournir des milliers de différentes adresses IP au point que nous nous devrions pouvoir attribuer un identifiant unique à chaque objet physique de la planète.

- **Deuxièmement**, les « objets » doivent communiquer entre eux. Grâce aux nouvelles technologies sans fil qui existe et qui rendent les communications possibles, telles que le Wifi, *LoRaWAN*, les communications à faible champ (Bluetooth), la communication en champ proche (NFC), la RFID, ainsi que les technologies ZigBee, Z-Wave.
- **Troisièmement**, afin d'obtenir des informations sur l'environnement, chaque « objet » doit avoir des capteurs. Il existe de nombreux capteurs notamment les capteurs de température, d'humidité, de lumière, de mouvement, de pression, infrarouge, à ultrasons, etc. Les nouveaux capteurs deviennent de plus en plus petits, économiques et durables.
- **Quatrièmement**, grâce au microcontrôleur que les « objets » doivent avoir, on peut gérer les capteurs, les communications ainsi que l'exécution des tâches. Il existe de nombreux microcontrôleurs pouvant être utilisé dans IdO selon le besoin.
- **Enfin**, afin de pouvoir exploiter la puissance de calcul, le stockage d'un serveur informatique, ainsi l'analyse et l'affichage des données, les services Cloud sont recommandés afin de pouvoir voir ce qui se passe et d'agir via des applications téléphoniques. Beaucoup de grandes entreprises y travaillent déjà, telles que Watson, IBM, la plate-forme Google Cloud, Azure, Microsoft, Oracle Cloud, etc.

La plupart de ces objets « objets IdO » seront également des systèmes embarqués et Internet des Objets est la façon dont ces appareils communiquent.

5. Les applications de l'IdO

L'IdO trouvera ses applications dans presque chaque aspect de notre vie. Ci-dessous quelques exemples (Perry Xiao, 2018) (Figure 1.2).



Figure 1.2 Les principaux domaines d'application de l'IdO.

5.1. La maison intelligente « Smart Home »

« Smart Home » ou une maison intelligente ou encore la domotique est un ensemble de techniques permettant de contrôler les appareils électriques utilisés dans les domiciles (éclairage, chauffage, audio-visuel, électro-ménager, etc.), d'automatiser et de programmer l'habitat. C'est un nouveau concept récent de plus en plus utilisé, il est issu de l'application à l'habitat de la programmation informatique, en profitant des technologies d'informatiques, des télécommunications, l'électronique. Toutes ces techniques et systèmes doivent être interopérables pour une gestion optimale. La domotique a pour rôle d'assurer des fonctions de sécurité, de gestion d'énergie, de contrôle des équipements (alarme, contrôle d'accès, température, éclairage, incendie...) et de communications, et en particulier de confort surtout pour les personnes âgées et les personnes handicapées.

L'IdO, nous permet de collecter et analyser des données provenant de ces personnes afin de diagnostiquer des maladies, d'évaluer les risques potentiels, d'identifier et de prévenir si possible les accidents tels que les chutes..., et de laisser les membres de la famille les surveiller à distance. Par conséquent, il est également possible de rapprocher davantage les personnes âgées et les personnes handicapées du monde extérieur et de briser les sentiments de solitude et d'isolement chez ces personnes ainsi de reconstruire une connexion avec le monde extérieur.

5.2. Soins de santé

L'IdO permet de gérer à distance les systèmes de santé et de notification d'urgence. Parmi les approches les plus populaires est l'utilisation des technologies portables. Ces appareils portables peuvent collecter une série de données « Health », telles que la fréquence cardiaque, la température corporelle et la pression artérielle, qui peuvent ensuite être transmises sans fil à un site distant pour être stockées et analysées. Ceci est également applicable pour diagnostiquer et traiter des patients à distance.

5.3. Transport

L'IdO peut révolutionner les systèmes de transport en améliorant la vie des gens. Par exemple avec les voitures connectées, organiser ses voyages deviendra plus facile, on peut éviter les embouteillages, de trouver une place de parking et de réduire les accidents de la circulation.

Les voitures sans conducteur auront sans aucun doute le plus gros impact. De nombreuses entreprises, telles que Tesla, Google, Uber, Volvo, Volkswagen, Audi et General Motors..., les développent et les promeuvent activement. Ces voitures intelligentes peuvent rendre notre trajet plus agréable et éventuellement beaucoup plus sûr.

L'IdO peut également améliorer la qualité des transports en commun. En connectant tous les panneaux d'informations et les panneaux publicitaires dans les gares et les aéroports, il aide les passagers à obtenir des mises à jour régulières et, en cas d'accident, à détecter rapidement les problèmes et à réduire les coûts de maintenance. En améliorant la visibilité de bout en bout, la gestion des entrepôts et la gestion de la flotte.

5.4. L'énergie

En intégrant des techniques de gestion, il est possible de réduire la consommation d'énergie de tous les appareils électriques par l'installation de capteurs et d'actionneurs. L'IdO modernisera également l'infrastructure du secteur d'énergie et par conséquent améliorer l'efficacité et la productivité.

5.5. L'environnement

Avec les différents capteurs environnementaux déployés, nous pouvons surveiller plus efficacement la qualité de l'air, la qualité de l'eau, les conditions du sol, les radiations et les produits chimiques dangereux. Nous pouvons également mieux prédire les séismes et les

tsunamis et détecter plus rapidement les incendies de forêt, les avalanches de neige et les glissements de terrain. Tout cela nous aidera à mieux protéger notre environnement. En marquant les animaux sauvages, en particulier les espèces menacées, nous pouvons étudier et mieux comprendre le comportement des animaux, et donc offrir de meilleures protections et des habitats plus sûrs. L'IDO permettra également de surveiller la qualité des sols et des cultures à distance et à n'importe quel moment de la journée, ce qui rend l'agriculture intelligente. Cela aura inévitablement un impact positif sur l'environnement.

6. Cloud Computing

6.1. Définition

L'idée du Cloud Computing n'est pas nouvelle, John McCarthy a déjà prédit que des installations informatiques seraient mises à la disposition du public à titre d'utilité, ce qui lui donnerait une similitude entre la téléphonie et l'informatique.

“ Le calcul peut un jour être organisé comme un service public, tout comme le système téléphonique public. Le service informatique pourrait devenir la base d'une nouvelle et importante industrie. ” John McCarthy, prenant la parole au MIT Centennial en 1961 [2].

Certes, il n'existe toujours pas de définition standard du Cloud Computing. Pour cette raison, des travaux ont récemment été faits sur la normalisation de la définition du Cloud Computing. À titre d'exemple, le travail de Yasmin Fathy et al. (2018), a comparé plus de 20 définitions différentes provenant de diverses sources pour confirmer une définition standard. Dans ce travail, nous adopterons la définition du Cloud Computing donnée par l'Institut National des Normes et de la Technologie (NIST), qui en couvre tous les aspects.

La définition NIST du Cloud Computing: *“le Cloud Computing est un modèle permettant un accès omniprésent, pratique et à la demande à un réseau partagé de ressources informatiques configurables (p. ex. réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement provisionnés et diffusés avec un minimum d'effort de gestion ou d'interaction avec les fournisseurs de services. ”* (Mell Peter et al, 2011).

6.2. Les types de Cloud Computing

Il existe différents types de nuages, chacun avec ses propres avantages et inconvénients (Zhan Qi et al,2010)

Cloud Public

Cloud dans lequel les fournisseurs de services fournissent leurs ressources sous forme de services au grand public. Les Clouds publics offrent plusieurs avantages clés aux fournisseurs de services, notamment l'absence d'investissement en capital initial dans l'infrastructure et le transfert des risques pour les fournisseurs d'infrastructure. Cependant, les Clouds publics manquent de contrôle précis sur les paramètres de données, de réseau et de sécurité, ce qui nuit à leur efficacité dans de nombreux scénarios commerciaux.

Cloud Privé

Également appelés nuages internes, les nuages privés sont conçus pour une utilisation exclusive par une organisation. Un nuage privé peut également être créé et géré par l'organisation ou par des fournisseurs externes. Un Cloud privé offre le meilleur degré de contrôle sur les performances, la fiabilité et la sécurité. Cependant, on leur reproche de ne pas leur offrir d'avantages tels que l'absence de coûts en capital initiaux.

Cloud Hybride

Un Cloud hybride est une combinaison de modèles de Cloud public et privé qui tente de gérer les restrictions de chaque approche. Dans un Cloud hybride, une partie de l'infrastructure de service exécute des Clouds privés tandis que la partie restante utilise des Clouds publics. Les nuages hybrides offrent une flexibilité supplémentaire par rapport aux nuages publics et privés. Plus précisément, ils offrent un contrôle et une sécurité plus stricts sur les données des applications par rapport aux Clouds publics, tout en facilitant l'expansion et la contraction des services à la demande. En revanche, la conception d'un Cloud hybride nécessite de déterminer avec soin la répartition la plus efficace entre les éléments de Cloud publics et privés

6.3. Les modèles du Cloud

En fonction du type de capacité fournie, il existe trois principaux scénarios d'utilisation des cloud (Jiyi Wu et al, 2010), comme indiqué dans la Figure 1.3 [3]

Infrastructure en tant que service (IaaS – « Infrastructure as a Service »)

Les IP gèrent un ensemble surdimensionné de ressources informatiques, telles que la capacité de stockage et de traitement. Grâce à la virtualisation, ils sont capables de scinder, d'affecter et de redimensionner dynamiquement ces ressources pour créer des systèmes ad hoc, à la demande des clients. Ils déploient les piles de systèmes logiciels qui exécutent leurs services.

C'est la situation de l'infrastructure en tant que service (IaaS)

Plate-forme en tant que service (PaaS– « Platform as a Service »)

Les systèmes en nuage offrent un niveau d'abstraction supplémentaire : au lieu de fournir une infrastructure virtualisée, ils peuvent fournir la plate-forme logicielle sur laquelle les systèmes s'exécutent. La taille des ressources matérielles requises par l'exécution des services est clairement définie. Ceci est noté en tant que plate-forme en tant que service (PaaS). Un exemple bien connu est le moteur Google Apps.

Logiciel en tant que Service (SaaS – « Software as a Service »)

Enfin, il existe des services d'intérêt potentiel pour un grand nombre d'utilisateurs hébergés dans des systèmes en nuage. Ceci est une alternative aux applications exécutées localement. Un exemple de ceci peut être les alternatives en ligne d'applications bureautiques typiques telles que les traitements de texte. Ce scénario est nommé logiciel en tant que service ou « Software as a Service » en anglais.

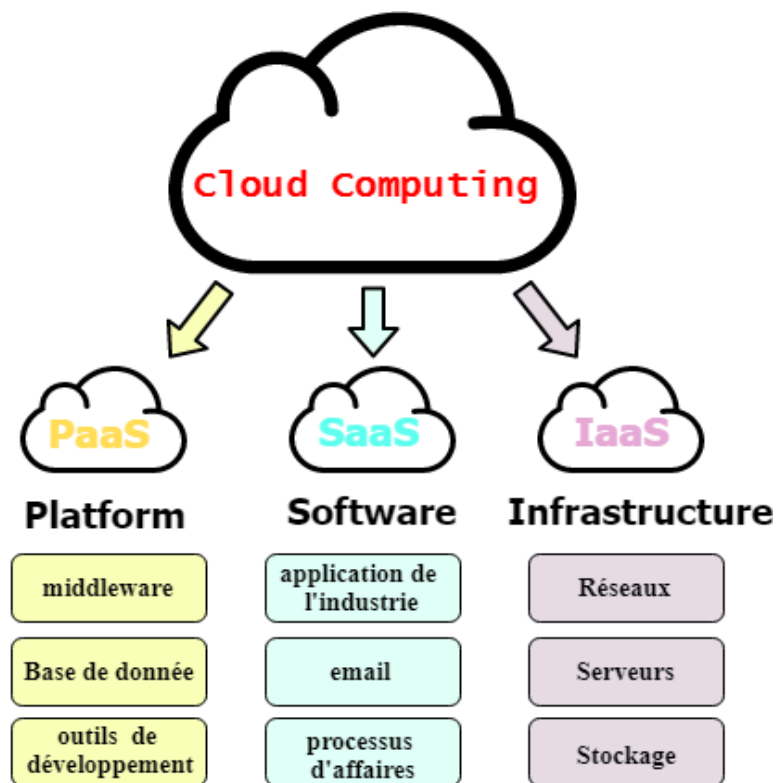


Figure 1.3 Les principaux modèles du Cloud

7. Conclusion

Dans ce chapitre, nous avons décrit les notions de base qui dépendent de notre projet. Tout d'abord que nous avons abordé une étude sur « l'Internet des Objets », leur domaine d'application ainsi son mode de fonctionnement et ceci pour entrer dans le vif du sujet. Enfin, nous avons présenté le Cloud Computing, ses types, ainsi que ses modèles.

Chapitre 2

L'apprentissage Automatique

1. Introduction

L'Internet des Objets permet d'interconnecté un ensemble de capteurs qui produisent de grandes quantités de données. Exploiter, traiter, analyser et extraire des informations utiles sur ces données est un défi qui commence à être relevé grâce aux capacités d'apprentissage automatique.

2. L'apprentissage automatique

2.1. Définition

Flach Peter (2012) a défini l'apprentissage automatique comme une étude systématique des algorithmes et des systèmes dans le but d'améliorer leurs performances ou bien leur connaissance avec l'expérience.

Mohammad Saeid Mahdavinejad et al. (2018) ont considéré l'apprentissage automatique comme un type de l'intelligence artificiel (IA) qui permet aux machines d'apprendre sans l'intervention de l'être humain.

De notre point de vue, nous constatons que l'apprentissage automatique est un ensemble d'algorithmes qui consiste à programmer les ordinateurs afin d'optimiser les critères de performance avec l'utilisation des exemples de données ou une expérience précédente afin de donner aux ordinateurs la capacité pour apprendre.

2.2. Les algorithmes d'apprentissage automatique

2.2.1 Le voisin K le plus proche (K- nearest neighbours)

KNN est une approche de classification supervisée utilisée dans le cadre de l'apprentissage automatique. Elle vise à classer chaque nouveau point de données avec l'utilisation d'une métrique de distance tel que la distance Euclidienne, Haming etc., tout en examinant les K points de données de l'apprentissage qui se rapprochent le plus de celui-ci dans l'entrée ou dans l'espace des fonctions (Figure 2.1.). L'inconvénient majeur de cette approche est qu'elle nécessite un grand espace de stockage pour un ensemble complet d'apprentissage, ce qui est impossible pour analyser des données volumineuses (Mohammad Saeid Mahdavinejad et al, 2018).

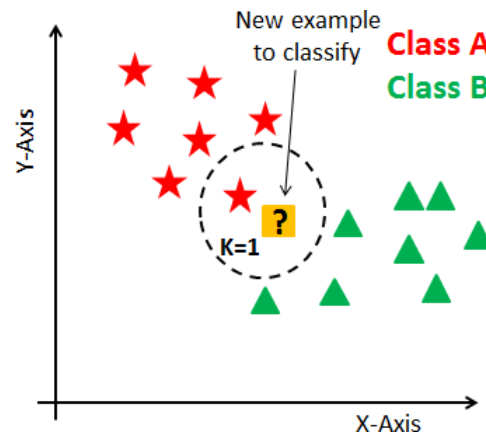


Figure 2.1 La classification avec KNN [4]

2.2.2. Machines à vecteurs de support (SVM)

SVM ou « Support Vector Machine » en anglais, est un algorithme d'apprentissage automatique supervisé souvent utilisé dans des problèmes de classification. Cet algorithme consiste à placer chaque donnée sous forme de points dans un espace de n dimension de caractéristiques et les classifier avec l'hyperplan qui différencie les classes. Elles sont très efficaces en termes de traitement de données en masse et de mémoire, grâce aux vecteurs de support pour la prédiction. Parmi ses limites est l'incapacité de déduire l'estimation de probabilité de manière directe [5].

2.2.3 Arbre de décision (Decision Tree)

L'arbre de décision est une branche importante de l'analyse de données. Elle a été utilisée dans divers domaines, notamment l'apprentissage automatique, la reconnaissance de forme, diagnostic médical, etc. L'arbre de décision possède plusieurs algorithmes qui génèrent une structure arborescente où chaque nœud interne désigne un test sur un attribut. Chaque branche représente le résultat du test et chaque nœud feuille contient une étiquette de classe (Figure 2.2.). Elle vise à partitionner récursivement l'espace d'attribut jusqu'à ce que tous les cas soient complètement partitionnés en sous-ensembles qui ne se chevauchent pas. Wu Tao et Qin Kun (2009). Parmi ses inconvénients est qu'elles sont instable et ont tendance à overfiter.

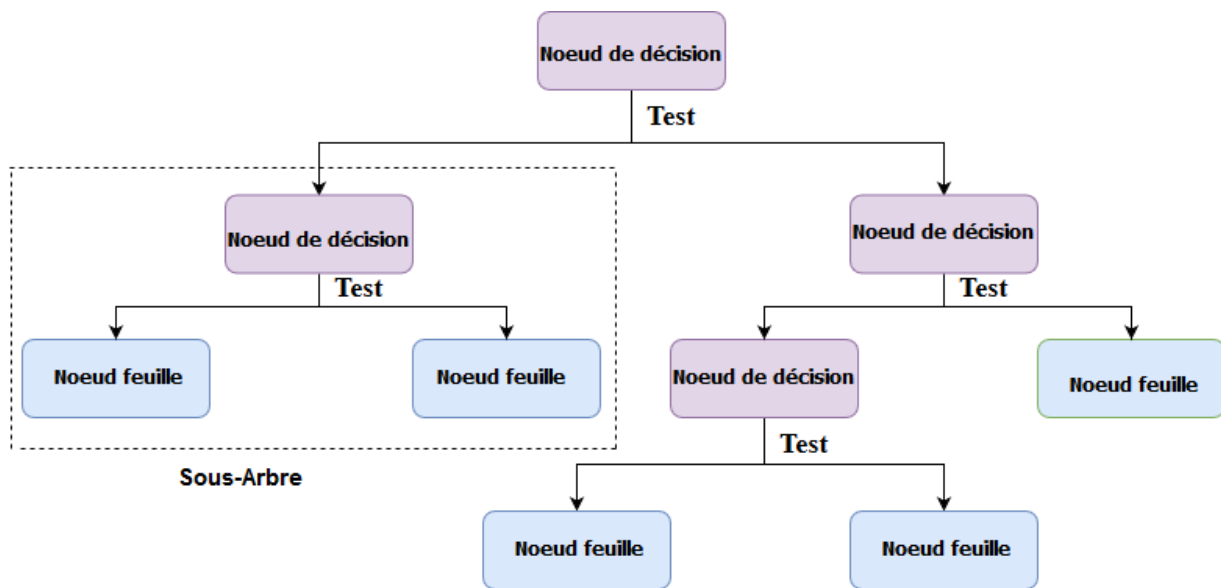


Figure 2.2 Architecture d'un arbre de décision

2.2.4 Forêts aléatoire

La forêt aléatoire, comme son nom l'indique, se compose d'un grand nombre d'arbres décisionnels individuels qui fonctionnent comme un ensemble. Chaque arbre de la forêt aléatoire crache une prédiction de classe, et la classe ayant obtenu le plus de votes devient la prédiction de notre modèle [6]. Cependant l'inconvénient majeur de cette approche est qu'elle est difficilement interprétable, difficilement améliorable et possède un entraînement plus lent

2.2.5 Réseaux de neurones

Les réseaux de neurones artificiels sont l'un des principaux outils utilisés dans l'apprentissage automatique. Ils sont des structures inspirées des circuits de neurones du système nerveux et sont composés d'unités informatiques interconnectées. Chaque neurone envoie et reçoit des impulsions d'autres neurones et ces changements sont modélisés sous forme de données. L'élément principal d'un réseau de neurones est le neurone artificiel (Moise et Netedu, 2012).

L'utilisation des réseaux de neurones offre deux avantages importants : ils permettent d'une part de calculer toute fonction calculable, et d'autre part ils sont particulièrement utiles pour le clustering, la classification, la reconnaissance des formes et le rapprochement de fonction (*mapping*) des problèmes (Bendiab, 2011). Malgré l'utilisation importante des réseaux de neurones, ils ont des inconvénients notamment la difficulté d'analyser et de comprendre le fonctionnement en face d'un problème donné (boîte noire), ainsi que la difficulté de choisir la structure la mieux adaptée aux problèmes (beaucoup de paramètres) La Figure 2.3 illustre une simple architecture d'un réseau de neurones avec une seule couche cachée.

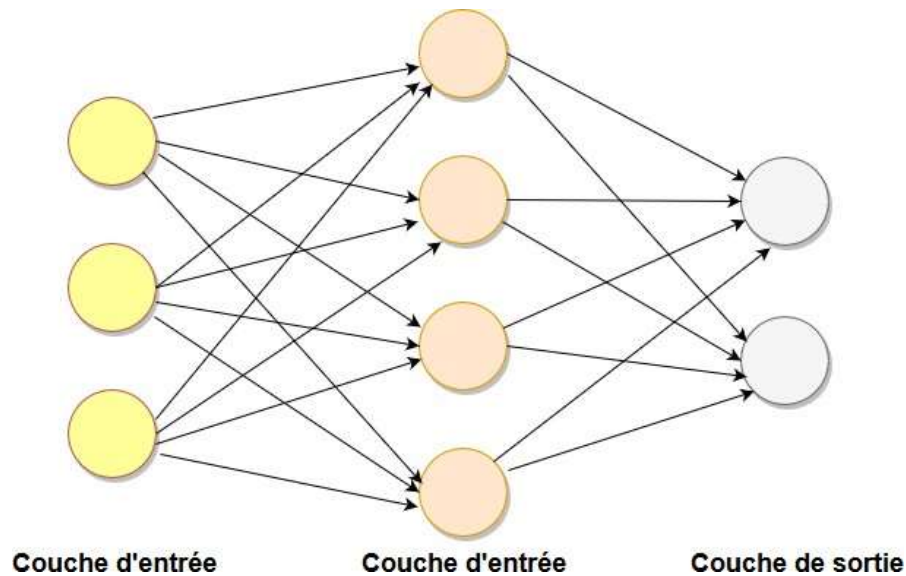


Figure 2.3 Architecture d'un réseau de neurones artificiel avec une seule couche cachée

3. L'utilisation des algorithmes d'apprentissage automatique dans l'IdO

Récemment, (Mojtaba Masoudinejad et al, 2018) ont présenté et évalué une nouvelle perspective pour estimer la localisation intérieure à l'aide d'une configuration logistique industrielle. Pour atteindre cet objectif ils ont utilisé plusieurs modèles d'apprentissage automatique à savoir le KNN et l'arbre de décision comme un outil pour l'analyse de données environnementales dans cette nouvelle perspective.

D'autres chercheurs, (Huimin Qian et al, 2010) ont proposé un Framework système qui permet la détection de localisation des personnes ainsi que la reconnaissance de leurs activités à partir des séquences vidéos. Un classifieur **SVM** a été utilisé dans ce travail pour reconnaître les activités humaines.

Shinde et Jayashree. (2017) ont proposé un système de surveillance sanitaire des animaux afin d'économiser l'effort et le temps de l'agriculteur. Ce système utilise la technologie de réseau de capteurs sans fil et des applications IoT pour collecter les informations de l'animal (température, fréquence cardiaque, cellule de charge, etc.). Ensuite le système stocke les informations dans une base de données, puis les compare avec les lectures initiales en utilisant la méthode de classification **Naïve Bayes**. Enfin, le système envoie des notifications à l'utilisateur si un changement radical survient dans les lectures.

Ani R et al. (2017) ont proposé un système de surveillance des patients pour les personnes affectées par un accident vasculaire cérébral (AVC) basé sur la méthode d'ensemble de classification qui est basée sur **les arbres-forets aléatoire**. Ce système a pour but de minimiser la récurrence future de cette maladie avec une alerte pour notifier le médecin et les aides-soignants de la variation des facteurs de risque de la maladie de l'AVC.

Elike Hodo et al. (2016) ont proposé une approche basée sur un **réseau de neurones** pour la détection d'intrusion sur un réseau IdO afin d'identifier les attaques de déni de service distribué DDoS / DOS. Cette approche s'est concentrée principalement sur la classification des schémas normaux et des schémas de menace sur un réseau IdO afin d'identifier les différents types d'attaques et à afficher les bonnes performances.

Algorithme d'apprentissage automatique	Traitement de données	Références
Le voisin le plus proche (KNN)	Classification	Mojtaba Masoudinejad et al. 2018
Bayes Naïve (NB)	Classification	Shinde et Jayashree.2017
Machines à vecteurs de support (SVM)	Classification	Huimin Qian et al. 2010
Arbres de décision (DT)	Classification/Régression	Masoudinejad et al. 2018
Forêt aléatoire (RF)	Classification/Régression	Ani R et al. 2017
Réseau de neurones artificiel (ANN)	Regression/Classification/Clustering/Feature extraction	Elike Hodo et al.2016
K-Means	Clustering	Ahod Alghuried.2017

Tableau 2.1 Travaux de recherche sur les algorithmes d'apprentissages automatique

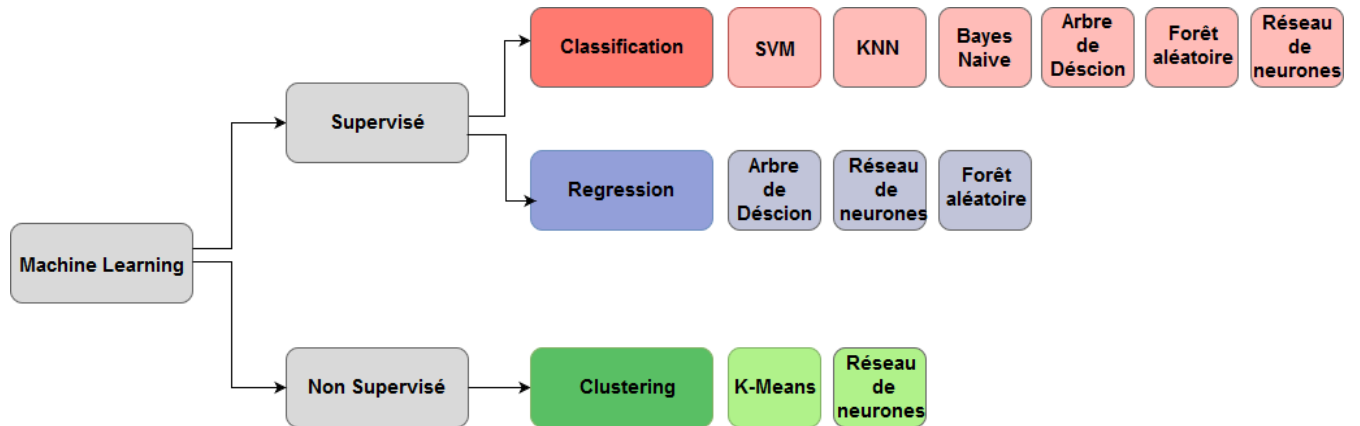


Figure 2.4 Taxonomie des algorithmes d'apprentissage automatique

4. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur l'apprentissage automatique. De nombreux algorithmes sont discutés. Sur la base de ces algorithmes nous avons présenté une taxonomie sur l'utilisation de ces algorithmes dans le domaine de l'Internet des Objets (IdO).

Chapitre 3

Systèmes Embarqués

1. Introduction

La révolution de l'Internet des Objets et ces équipements (capteurs, actionneurs, processeurs, etc.) impliquent la conception et le développement des cartes matériels et des systèmes logiciels, d'où la naissance des systèmes embarqués.

Un système embarqué connecté permet aux technologies de se connecter sur plusieurs appareils, plates-formes et réseaux, créant ainsi un réseau de communication qui révolutionne la manière dont nous interagissons numériquement avec le monde [7].

2. Système Embarqué

Selon Perry Xiao (2018), il s'appelle embarqué parce que le système informatique est embarqué à un périphérique matériel. Il est caractérisé par un faible cout, faible consommation d'énergie ainsi que par sa taille petite. Son importance incarné dans son utilisation dans de nombreux appareils quotidiens tels que les voitures, les appareils photo, les téléviseurs intelligents, les montres numériques, et machines à laver. Il est conçu pour effectuer des tâches spécifiques et souvent un système en temps réel.

2.1. Architecture d'un système embarqué

Selon Perry Xiao (2018), un système embarqué est constitué des dispositifs montré dans la figure 3.1

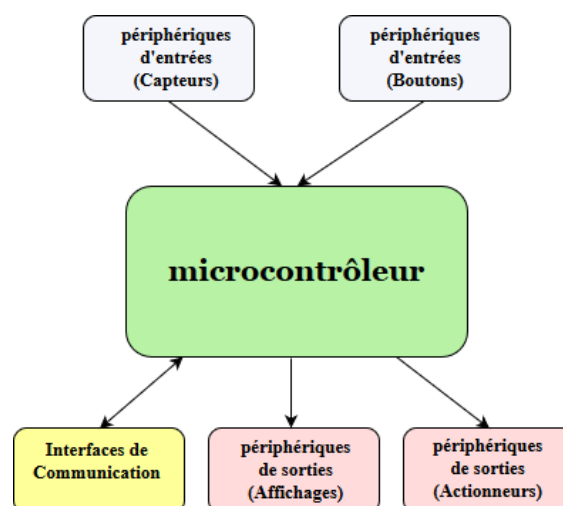


Figure 3.1 Schéma d'un système embarqué typique

Nous donnons dans ce qui suit la description de chaque partie (Perry Xiao, 2018).

- **Microcontrôleur**, c'est le noyau d'un système embarqué qui va prendre en charge la partie traitement des informations et envoyer des ordres. C'est lui qui va exécuter le programme embarqué dans le microcontrôleur. De plus, il est doté d'une mémoire et de tous les périphériques d'entrée/sortie.
- **Les entrées**, sont deux types numérique ou analogique ils sont généralement utilisés pour que le système embarqué puisse interagir avec le monde extérieure à travers la lecture des données de capteurs exp : capteur de température, ou d'autres types de périphériques d'entrée exp : boutons, etc.
- **Les sorties**, ont également les mêmes types que les entrées ainsi le même objectif qui permet d'interagir avec le monde extérieur. Ils sont utilisés pour l'affichage ou bien pour autres dispositifs exp : actionneurs.
- **Les interfaces de communication**, ont pour objectif d'assurer le lien entre un système embarqué et les autres périphériques, elles peuvent être filaire ou sans fil. Parmi ces interfaces, on note : Ethernet, USB, infrarouge, ZigBee, Wifi et Bluetooth.

3. Carte d'interface

Les cartes d'interfaces du système (Arduino, MiniZed, Rosebery, Dragon, Orange, etc.) sont des familles de microcontrôleur qui peuvent capter et contrôler d'avantage des choses du monde matériel. Ce sont des plateformes d'électronique programmée généralement open-source.

Un microcontrôleur est le composant principal dans ces cartes. Il est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte et mémoire vive), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un gigahertz) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels [8].

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés [8].

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués inclus dans les cartes électroniques.

4. Exemples de cartes d'interface

Il existe une variété de carte d'interface utilisé dans beaucoup de domaines y compris les domaines de l'IdO. Chaque carte a des avantages et des inconvénients, chaque concepteur peut choisir la carte qui le convient le plus. Le Tableau 3.1 énumère quelque uns.

Carte d'interface	Caractéristique
Arduino	Mini automate fait pour exécuter de la mono-tâche. Une fois qu'on charge un programme, il devient autonome et l'exécute en boucle. Elle est idéale pour commander des volets roulant, régulation de chaudière, gestion d'aquarium, pilotage de moteurs... [9]
Raspberry Pi	Raspberry Pi est plus qu'une carte, elle est considérée comme mini PC doté d'un système d'exploitation, donc peut faire tourner plusieurs programmes. Les interactions avec le Raspi sont plus faciles. Il est plus proche de l'informatique que l'Arduino. Idéal pour utiliser comme serveur, surveillance vidéo (domotique), passerelle... [9]
Banana Pi	Elle est plus qu'une carte. Elle est considérée comme mini PC à carte unique. Elle supporte Android, Ubuntu, ArchLinux, Fedora, Debian et Gentoo Linux. Le Banana Pi peut être utilisé comme plateforme de développement de logiciel open-source ainsi que beaucoup d'autres applications. Elle dispose de connecteurs pour la carte SD, des ports USB 2.0, de prises HDMI, un port Gigabit Ethernet, un récepteur infrarouge, un connecteur GPIO à 26 broches, etc. [10]
Orange Pi	Est une copie chinoise de la Raspberry Pi, à un coût nettement inférieur et avec de meilleures caractéristiques, sur le papier tout s'annonce donc plutôt bien.[11]
Minized	La carte MINIZED d'Avnet est une carte de développement à coeur MiniZed Zynq 7Z007S. Il cible les développeurs Zynq d'entrée de gamme avec une plate-forme de prototypage faible coût. Cette conception compacte offre une connectivité embarquée via USB, Wi-Fi et Bluetooth. MiniZed fournit une conception de référence matérielle efficace, alors qu'il s'agit également d'une carte peut coûteuse qui peut être utilisée pour organiser des ateliers et des tutoriels. Son objectif est de mettre en avant la puissance de Zynq, où le cœur du processeur Cortex-A9™ s'intègre de manière transparente à la structure programmable. [12]
....

Tableau 3.1 Les cartes d'interface les plus connu

Dans notre projet nous utilisons la carte Minized comme interface entre les capteurs et le cloud. Pour faire la conception et la programmation de cette carte, nous utilisons les logiciels Vivado et le SDK de Xilinx.

5. Famille de la carte Minized

Actuellement, il existe plusieurs versions de ce module. Nous citons à titre d'exemple les suivantes [13] :

- **MicroZed** : carte de développement à faible coût basée sur le système sur puce (Xilinx Zynq-7000 AP). Elle a une conception unique qui permet d'être utilisée comme une carte d'évaluation autonome.
- **PicoZed** : très flexible et robuste basé sur le même système sur puce da MicroZed. Ce module contient les fonctions communes requises pour prendre en charge le cœur de la plupart des conceptions de SoC (Système sur puce).
- **ZedBoard** : contient toutes les interfaces nécessaires et les fonctions de support permettant une large gamme d'applications. Les fonctionnalités d'extension de cette carte la rendent idéale pour le prototypage rapide et le développement de la validation de principe.

Parmi ces types, nous avons choisi la carte Minized. L'intérêt principal de cette carte est de faciliter la mise en œuvre de notre système qui sera détaillée par la suite

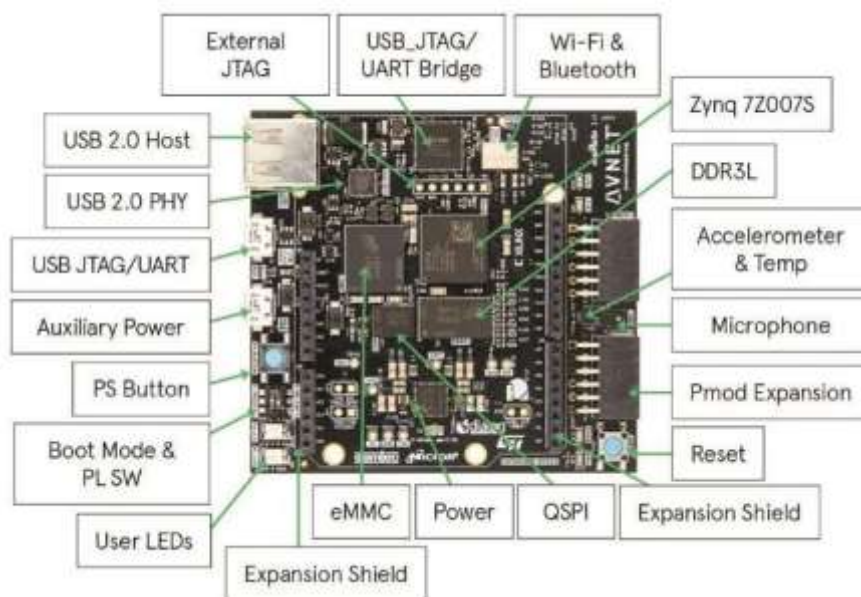


Figure 3.2 La carte Minized [14]

6. Conclusion

Dans ce dernier chapitre de l'état de l'art, nous avons présenté d'autres notions qui vont nous aider durant le développement de notre projet. Tout d'abord, on a fait une description détaillée sur le système embarqué et son architecture. Ensuite nous avons décrit les cartes d'interfaces, citer quelques exemples de la carte les plus utilisés et choisir par la fin celle la plus appropriée à notre projet.

Ce chapitre ainsi que les chapitres précédents relèvent ainsi une utilité majeure pour ce qui suit puisqu'il détaille des notions exploitées au sein de la partie réalisation de notre projet.

Chapitre 4

Étude Conceptuelle

1. Introduction

Après avoir défini les concepts théoriques de base sur l'IdO, l'apprentissage automatique et les systèmes embarqués dans les trois premiers chapitres, nous passons à la deuxième partie de notre travail qui consiste à présenter progressivement la conception et l'implémentation du système. Dans ce chapitre nous présentons dans un premier lieu l'objectif du projet ainsi son architecture générale où les différentes grandes parties sont clairement identifiées. Ensuite, nous détaillons la conception de chaque partie tout en mettant en claire à la fois l'aspect statique et dynamique de futur système.

2. Objectif du projet et méthodologie

Afin de bénéficier des avantages des applications IdO nous voulons réaliser une plateforme IdO, et utiliser l'apprentissage automatique pour prédire la localisation intérieure comme exemple d'application.

La localisation intérieure est en train de devenir un nouveau paradigme dans l'internet des objets. Cela est dû à la limite des GPS dans les environnements intérieurs à savoir la mauvaise pénétration de ses signaux à l'intérieur des immeubles, ainsi le cout considérablement cher de l'intégration des récepteur GPS dans chaque objet.

Le système à concevoir permet d'interconnecter un ensemble de capteurs, de recueillir et fusionner un ensemble des informations issues de ces capteurs afin que le système devienne capable d'utiliser ces données. Pour cela on a construit une plateforme qui est composée de capteurs de température et d'humidité, une carte contrôleur Minized où notre système est embarqué et une carte Transmetteur/récepteur pour transmettre et recevoir les données.

Ce système fait la capture de données brute issue de capteurs, les transmet vers le cloud avec un protocole de l'internet des objets « LoRaWAN » via une passerelle IoT ainsi le stockage de ces données dans le cloud. Ce dernier comporte une interface où on pourra visualiser le changement de ces données.

Le deuxième objectif est d'extraire ces données du cloud, les normaliser afin d'appliquer quelques algorithmes d'apprentissage automatique et d'apprentissage profond notamment les machines à vecteurs de support (SVM), les arbres de décision (DT), les classificateurs de forêt aléatoire (RFC), le voisin K le plus proche (KNN) et les réseaux de neurones artificiels(ANN) dans le but de prédire la localisation intérieure.

3. Méthode de conception adoptée

Les méthodes de conception définissent une représentation souvent graphique qui permet d'une part de manipuler aisément les modèles, et d'autre part de communiquer et d'échanger l'information entre les différents intervenants.

Une bonne représentation recherche l'équilibre entre la densité d'information et la lisibilité.

Pour la représentation des différents modèles issues de la conception nous choisissons les diagrammes UML. Car ceci nous sera très utile pour exprimer les différents modèles, séquençement et interactions entre les composants de notre système de façon claire.

UML (Unified Modeling Language)

UML est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, à spécifier et documenter des systèmes, à esquisser des architectures logicielles. Il s'articule autour de plusieurs types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel mais, on va représenter seulement ceux qui sont utilisés dans notre projet et qui sont :

- *Les diagrammes de séquence* : représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation.
- *Les diagrammes d'activité* : sont des diagrammes comportementaux d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système.

4. Architecture générale du système

4.1. Interfaces du système

Le système que nous venons de réaliser est composé de trois parties principales (Figure 4.1).

- *Interface physique pour la capture des données* : c'est un sous-système composé d'une carte Minized (ou notre système est embarqué) et des capteurs. Il permet de collecter des informations sur leur environnement comme la variation de température via une sonde thermique.

Ce sous-système sert aussi de pont entre les capteurs, le module de contrôle et le cloud.

- *Interface physique pour l'IdO* : c'est un autre sous-système qui est composé d'une carte Transmetteur/Récepteur, une passerelle IdO et un cloud .Ce sous-système a pour

rôle de transmettre les données collectées par le premier sous-système vers le cloud à travers le protocole LoRaWAN.

- **Interface d'apprentissage automatique** : c'est un ensemble d'algorithmes d'apprentissage automatique qui sont appliqués sur les données capturées et traitées par les sous-systèmes précédents dans le but d'avoir une localisation prédite avec une meilleure précision.

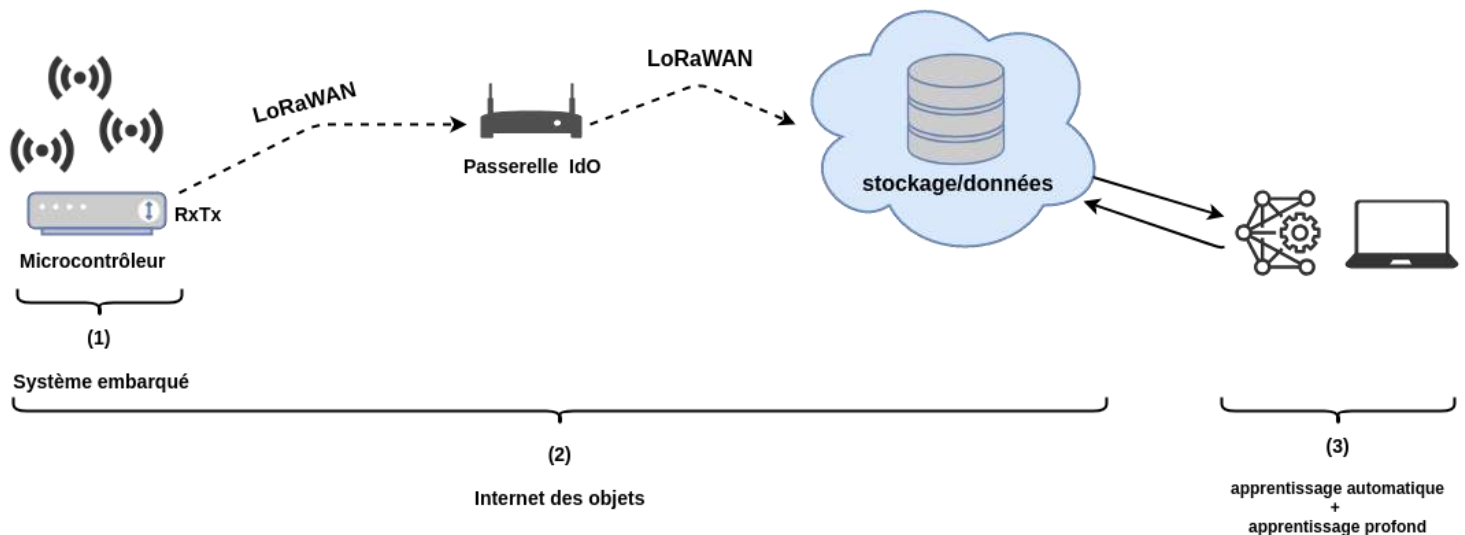


Figure 4.1 Architecture générale du système

4.2. Composants physique

4.2.1 Capteurs

Un capteur est un appareil qui détecte et répond à un type d'entrée de l'environnement physique. L'entrée spécifique peut être la lumière, la chaleur, le mouvement, l'humidité, la pression ou l'un des nombreux autres phénomènes environnementaux. La sortie est généralement un signal qui est converti en affichage lisible par l'homme [15]. Ces capteurs envoient leurs sorties aux microcontrôleurs.

4.2.2 Microcontrôleurs

Un microcontrôleur est un système autonome avec des périphériques, une mémoire et un processeur pouvant être utilisé comme système embarqué. Certains d'entre eux sont plus sophistiqués, tandis que d'autres ont des exigences minimales en termes de longueur de mémoire et de programmation et une faible complexité logicielle. [16]

Ces microcontrôleurs envoient leur sortie à un périphérique qui lit leur sortie et la traite. Dans notre plateforme, le microcontrôleur lit les données des capteurs et les envoient au cloud, ou elles sont stockées.

4.2.3 Passerelle

Une passerelle Internet des Objets permet à un appareil de transmettre des données à l'aide de ses capteurs à un emplacement distant [17].

Chaque microcontrôleur peut être connecté à la même passerelle(Gateway) formant un réseau en étoile, cette passerelle collecte les données de tous les microcontrôleurs, les filtre (en fonction de la priorité du message) et les envoie périodiquement au cloud.

4.2.4 Carte Transmetteur/Recepteur (Transceiver)

Abréviation de « émetteur-récepteur », une carte qui transmet et reçoit des signaux analogiques ou numériques. Dans les communications radio, un émetteur-récepteur est une radio bidirectionnelle qui combine un émetteur radio et un récepteur qui échange des informations en mode semi-duplex. [18]

4.3. Déroulement et fonctionnement

Le fonctionnement du système est basé sur l'interaction de ses composants. Ceux-ci échangent d'une manière continue, dynamique et autonome différentes informations.

La *figure 4.2* présent le diagramme de séquence décrivant cette interaction :

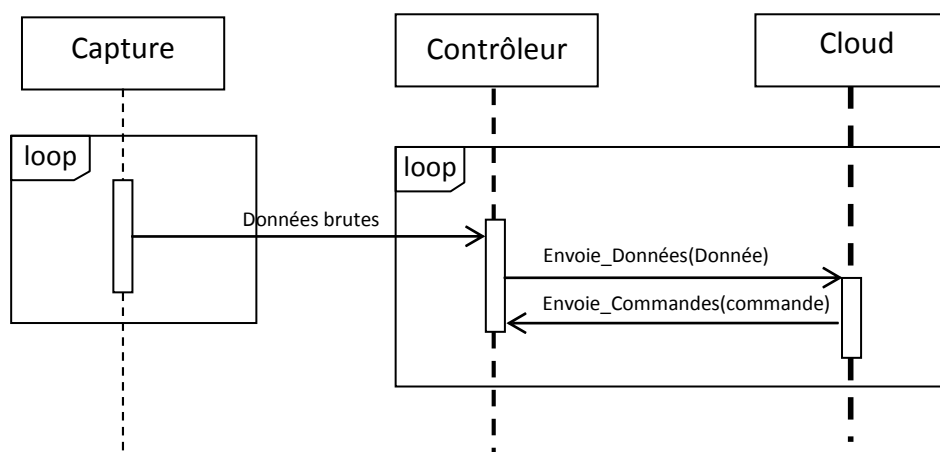


Figure 4.2 Diagramme de séquence du fonctionnement du système

- Le sous-système de capture fonctionne en permanence, il recueille des données brutes sur l'environnement à partir des capteurs et les envoie, avec certaine fréquence, au module de contrôle et de commande.
- Le module de contrôle reçoit les données brutes issues du sous-système de capture, traite ses données, puis les envoie au cloud.
- Dans le cloud Les données peuvent être affichées (à un utilisateur) dans son interface graphique, ainsi l'utilisateur peut envoyer des commandes au microcontrôleur à partir de cette interface. Les commandes reçues sont traduites en des actions (exemple envoyer la température chaque 10 secondes). La Figure 4.3 illustre le digramme de séquence détaillé.

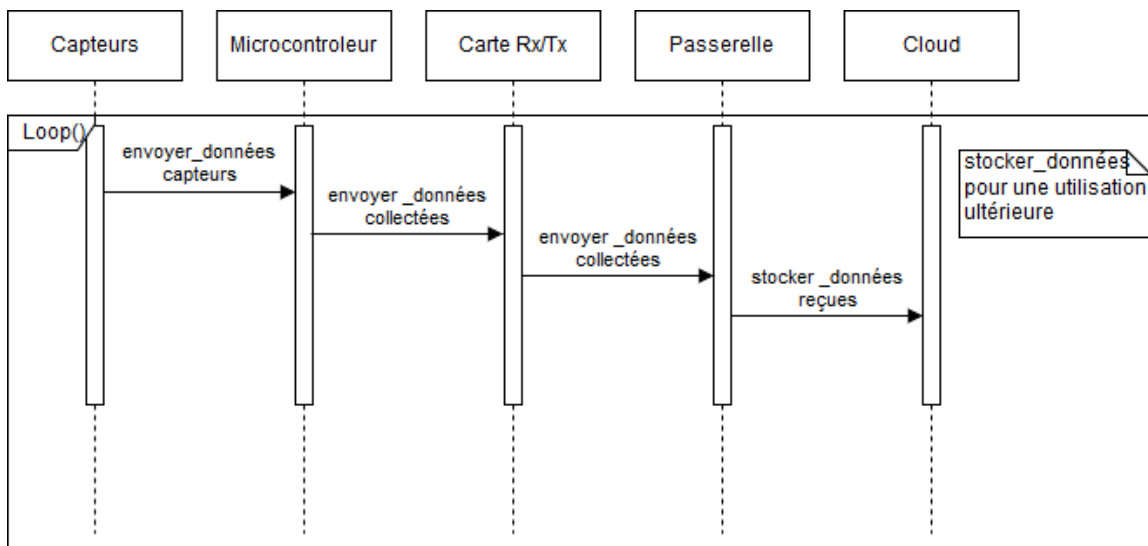


Figure 4.3 Diagramme de séquence correspond au scénario détaillé d'automatisation

5. Configuration physique de l'endroit où on a fait l'expérimentation

L'endroit visé dans notre projet a été l'université de Québec à Trois Rivières (Canada) et plus précisément le pavillon Provencher. Dans notre expérimentation on a choisie trois endroits différents, on a collecté les données de chacun de ces endroits (des milliers de données).

La Figure 4.4 illustre le plan de cet endroit ainsi que le plan des endroits où les mesures sont faites.



■ Correspond à l'emplacement A

■ Correspond à l'emplacement C

■ Correspond à l'emplacement B

■ Correspond au routeur

Figure 4.4 Plan de l'expérimentation

6. Principe de reconnaissance du contexte de localisation

Comme il n'y a, pratiquement, aucun capteur qui peut identifier d'une manière directe la localisation intérieurs sauf le GPS qui malheureusement ne fonctionne pas bien à l'intérieur, nous présentons dans ce qui suit comment les données brutes fournies par le sous- système de capture peuvent être traitées et fusionnées pour reconnaître la localisation.

La localisation est prédite à travers la fusion de différentes données (données de signaux et données environnementales) la Figure 4.5 illustre ce principe

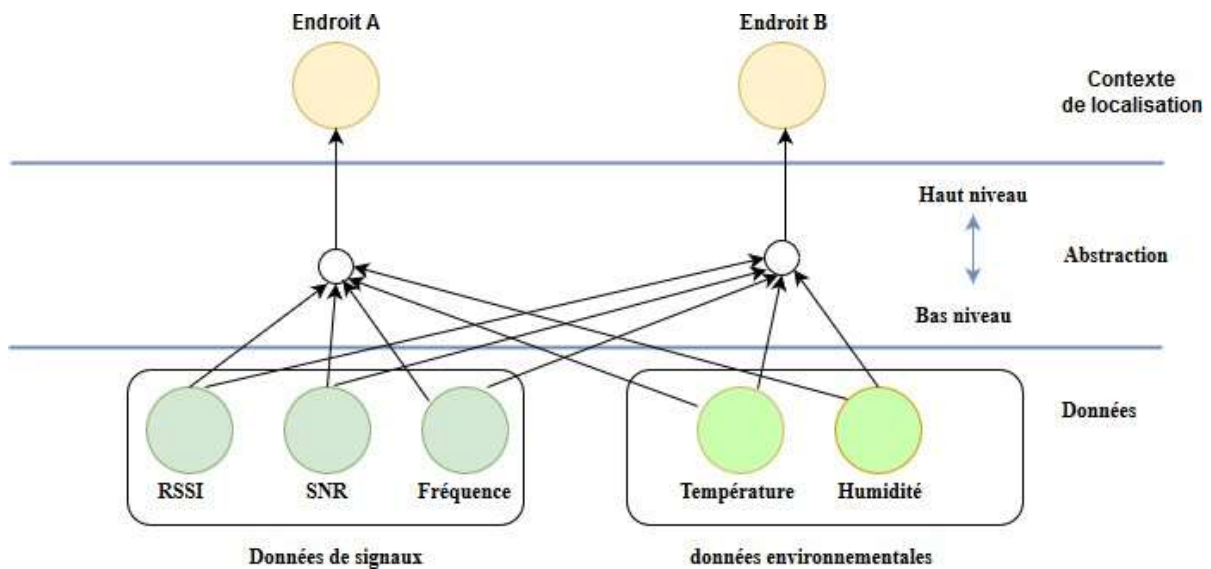


Figure 4.5 Représentation de la relation entre les données et le contexte de localisation

7. Architecture détaillé des composants du système

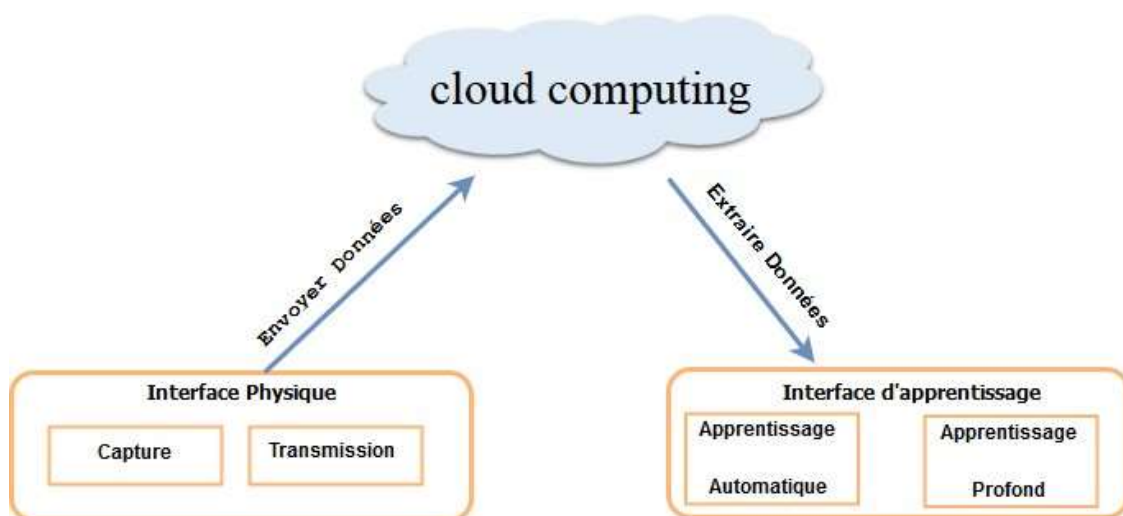


Figure 4.6 Architecture détaillé du système

7.1. Interface physique

L'interface physique inclut le sous-système de capture et le sous-système de transmission de données. Cette interface est ainsi dite car c'est celle qui permet à un tel système d'interagir directement avec son environnement physique. Elle est composée des éléments matériels (les

microcontrôleurs, composants électroniques) et des programmes embarqués dédiés pour leur fonctionnement. Dans ce qui suit nous détaillons chacun de ces sous-systèmes.

7.1.1. Interface pour la capture des données brutes du contexte

L'interface ou le sous-système physique de capture composé d'une carte Minized, de tous les capteurs, et ainsi que les composants électroniques de raccordement. Ce sous-système fonctionne grâce à une application embarquée dans la carte Minized.

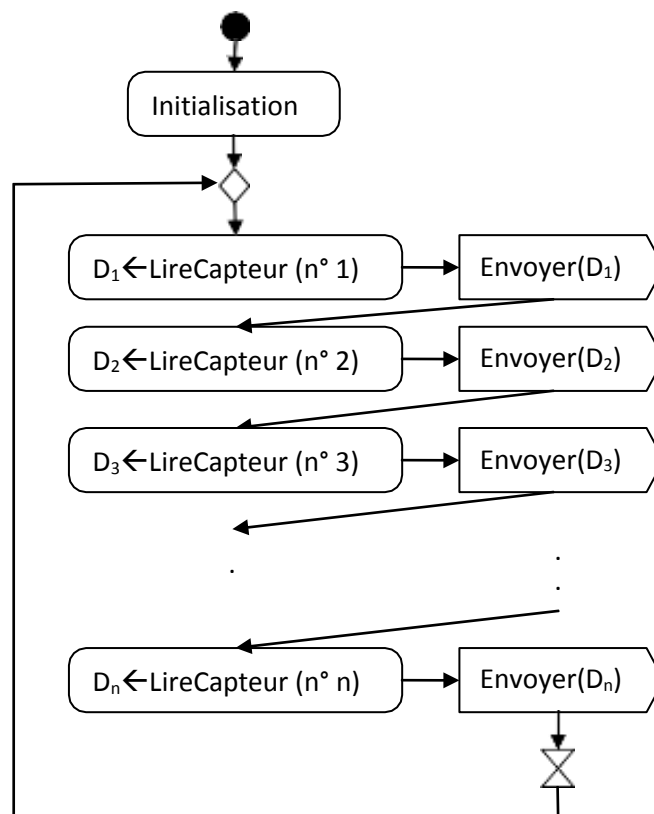


Figure 4.7 Diagramme d'activité correspond au sous-système capture

La **Figure 4.7** illustre le diagramme d'activité correspond au sous-système de capture des données brutes de contexte modélisant le traitement de l'application embarquée à être exécutée dans la Minized. Les deux tâches principales dans l'application embarquée sont : $D_i \leftarrow \text{LireCapteur}(n^\circ i)$ et $\text{Envoyer}(D_i)$ qui consistent respectivement en la lecture de l'information fournie par le capteur numéro i et en l'envoi de cette information au module de contrôle et commande.

7.1.2. Interface IdO pour la transmission

Le deuxième sous-système de l'interface physique est celui à utiliser pour Transmettre des informations sur l'environnement. Celui-ci est composé d'une carte Rx/Tx et d'une passerelle qui sert d'intermédiaire entre la carte et le cloud pour transmettre les différents types de donnée. Le sous-système action permet au système d'agir directement sur le monde physique. Il fonctionne grâce à une application embarquée (s'exécutant sur la carte Minized).

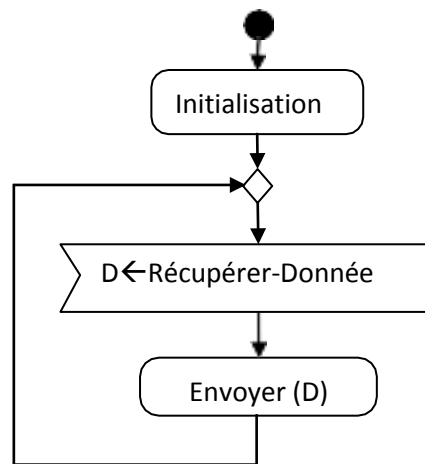


Figure 4.8 Diagramme d'activité correspond au sous-système de transmission

7.1.3. Scénario d'application

Nous avons implémenté notre application pour permettre à notre carte Minized de communiquer avec la carte Rx/Tx ainsi avec le cloud pour envoyer et recevoir les données via une passerelle tout en passant avec une configuration de cette passerelle.

Pour que cela fonctionne nous avons importé une bibliothèque « SymphonyLink, », configuré la passerelle et associé un jeton d'application au module pour qu'il puisse communiquer avec le cloud, ce module doit s'enregistrer en lui informant de son jeton d'application. L'enregistrement du module est exécuté en transmettant le jeton par liaison radio à l'aide de notre librairie. Puis nous pouvons ajouter un jeton de réseau pour séparer les réseaux Symphony distinct.

Un module et une passerelle ne peuvent pas communiquer sauf s'ils publient tous deux le même jeton de réseau. *La figure 4.9* illustre le fonctionnement de notre application avec tous les états possibles.

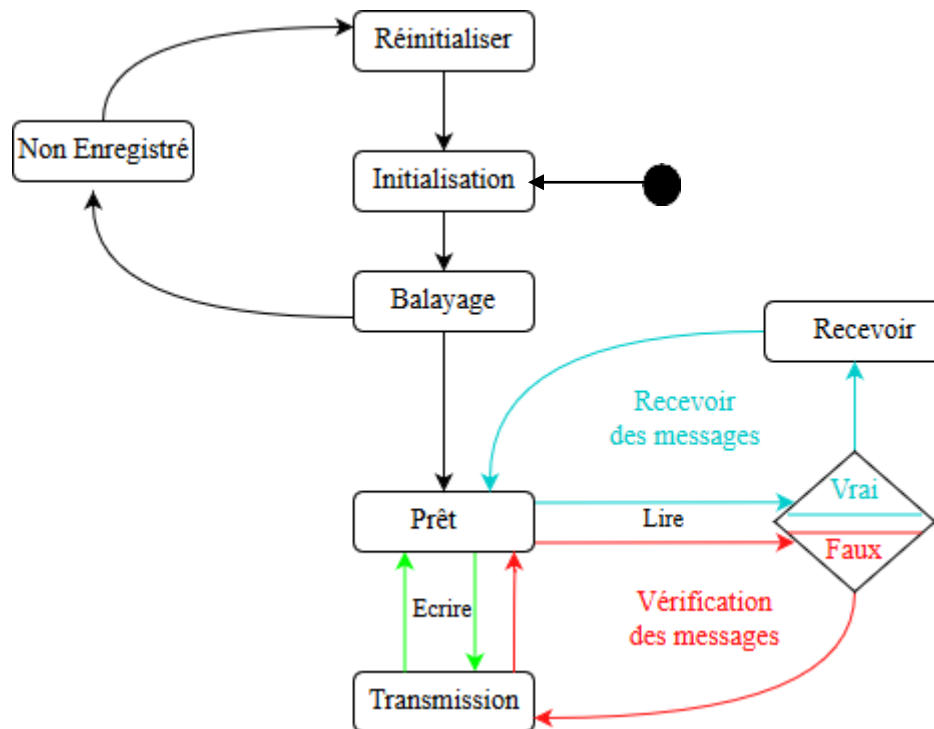


Figure 4.9 Déroulement de l'application

7.2. Interface d'apprentissage

La technique de localisation en intérieur est une technologie clé pour le paradigme de l'Internet des Objets (IdO) du futur. L'amélioration de la précision de la localisation intérieure élargira l'horizon des applications IdO intérieures. (Yen-Kai Cheng et al, 2016). Afin d'appliquer un apprentissage sur les données, un script python est utilisé pour extraire ces données à partir du cloud.

7.2.1. L'apprentissage automatique

Cette partie consiste à appliquer quelques algorithmes d'apprentissage automatique dans le but d'avoir une meilleure précision pour la prédiction de la localisation en intérieur. Avant d'appliquer ces algorithmes un prétraitement sur les données est nécessaire. Vu que ces données proviennent avec des ordres de grandeurs différents, cette différence d'échelle peut conduire à des performances moindres.

Pour pallier à cela et pour améliorer les performances, des traitements préparatoires sur les données existent. Notamment la normalisation des données (**Min-Max Normalization**).

A l'issue de cette transformation, les données seront comprises dans un intervalle fixe [0,1]. Le but d'avoir un tel intervalle restreint est de réduire l'espace de variation des valeurs des données.

Cette normalisation est effectuée par la formule de la technique du **Min-Max Normalization** :

$$X_{normalise} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Ou :

- **X_{min} : La plus petite valeur observée par la colonne X (Feature X).**
- **X_{max} : La plus grande valeur observée par la colonne X (Feature X).**
- **X : La valeur de la colonne (Feature) qu'on cherche à normaliser.**

Afin d'exécuter les algorithmes d'apprentissage automatique nous avons utilisé la boîte à outils « Classification learner » de Matlab.

K-plus proche voisin (K-NN)

- Choisir le nombre **K** de voisins.
- Prendre les **K** voisins les plus proches du nouveau point de données, en fonction de la distance euclidienne entre les voisins.
- Parmi ces **K** voisins, Compter le nombre de points de données dans chaque catégorie.
- Assigner le nouveau point de données à la catégorie où le plus de voisins est compté.

Machine à vecteurs de support (SVM)

- Préparer la matrice de motif.
- Sélectionner la fonction du noyau à utiliser.
- Sélectionnez le paramètre de la fonction du noyau.
- Exécuter l'algorithme d'apprentissage et obtenir le **α** .
- Les données invisibles peuvent être classées à l'aide du **α** et du vecteur support.

Arbre de décision

- Prétraiter les données du moteur de croissance de l'arbre de décision.
- Définir les caractéristiques de modélisation d'entrée et de sortie.
- Sélectionner le paramètre de croissance de l'arbre de décision.
- Cluster et traiter chaque branche formant un champ d'entrée.
- Sélectionner la branche de l'arbre de décision candidat.
- Compléter la forme et le contenu de l'arbre de décision final.

Forêt aléatoire

- Choisir au hasard N points de données de l'ensemble de formation.
- Construire l'arbre de décision associé à ces N points de données.
- Choisir le nombre d'arbres **Narbres** à construire puis la répétition des étapes (1,2)
- Pour un nouveau point de données, indiquer à chacun des arbres **Narbres** la valeur de Y pour le point de données en question.
- Assigner au nouveau point de données la moyenne sur toute la valeur Y prédite.

7.2.2. L'apprentissage profond

L'apprentissage profond s'appuie sur les réseaux de neurones artificiels ou les résultats d'une couche de « neurones » sert d'entrée aux calculs d'une autre couche et ainsi de suite, dans ce qui suit les composants de notre réseau de neurones :

- ✓ Une **couche d'entrée** qui contient : Température, Rssi, SNR, Fréquence comme des données d'entrée.
- ✓ Une **couche cachées** qui capture un ensemble d'entrées pondérées et produit une sortie via une fonction d'activation.
- ✓ . Une **couche de sortie** qui contient l'endroit À prédire (exp : endroit A, endroit B...)
- ✓ Une fonction d'activation **softmax** pour une classification multi-class

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ pour tout } j \in \{1, \dots, K\}.$$
- ✓ Une fonction d'optimisation **Adam**.
- ✓ Un ensemble de **poids(w)** entre chaque couche,
- ✓ Une fonction de perte **lossfunction** pour calculer l'erreur de classification.

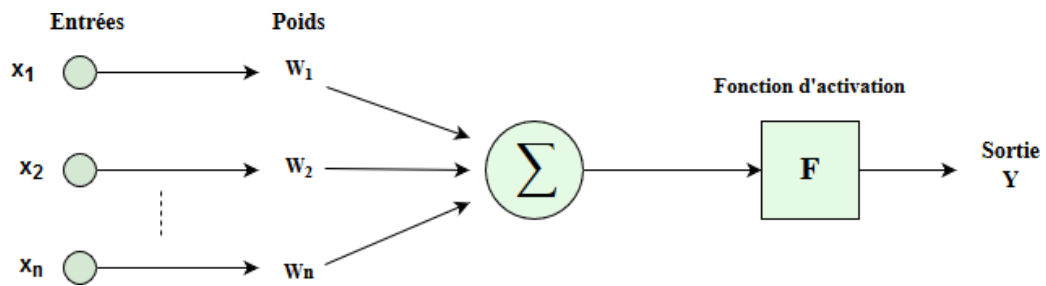


Figure 4.10 Schéma du fonctionnement d'un neurone [19]

8. Conclusion

Tout au long de ce chapitre, nous avons exposé les différentes fonctionnalités auxquels doit répondre le système à réaliser ainsi que la solution que nous avons adoptée pour la satisfaction de ces exigences stipulées. Dans le chapitre suivant, nous allons aborder l'implémentation de cette solution proposée tout en évoquant l'ensemble des choix techniques et logiciels qui nous ont aidés dans la réalisation de cette tâche.

Chapitre 5

Implémentation

1. Introduction

Le travail à réaliser comporte deux parties, une partie théorique, où les notions et les concepts de base du thème sont présentés, et une partie pratique où les différentes implémentations requises sont détaillées. Dans le chapitre précédent, nous avons présenté les composants et les idées que nous allons mettre en place d'un point de vue conceptuel. Dans ce chapitre, nous exposerons l'implémentation et la réalisation de notre système. Nous présentons également les différents outils et les technologies utilisés pour le développer. Enfin, nous présentons et expliquons les différents résultats acquis.

2. Implémentation de l'interface physique de capture

L'interface physique de capture est le sous-système composé d'une carte Minized où notre système est embarqué, et des capteurs. Il sert ainsi de pont entre les capteurs, le module de contrôle et de commande et le cloud. Dans cette section, nous expliquons l'implémentation de cette interface. La figure 5.1 illustre la conception matérielle interne de notre module de contrôle fait avec le logiciel Vivado.

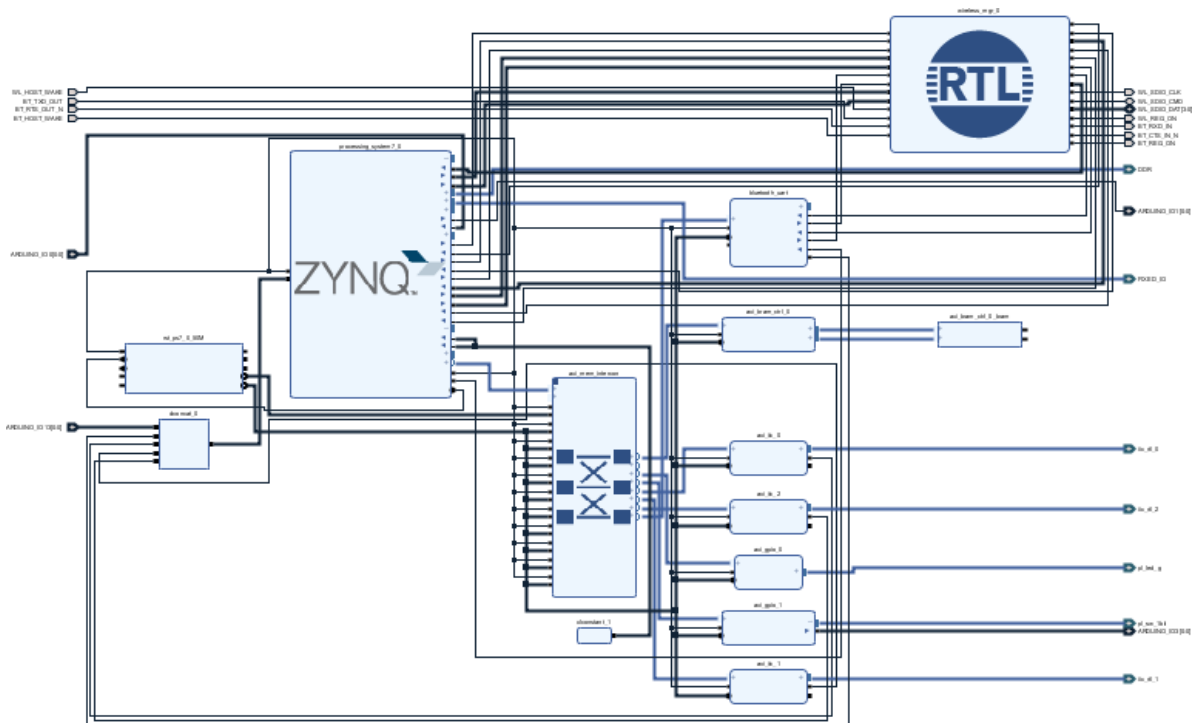


Figure 5.1 Conception matériel du microcontrôleur

2.1. Les composants principaux

Zynq

Zynq-7000 est basée sur l'architecture système sur puce (SoC). Ces produits intègrent un système de traitement (PS) à double cœur ARM® Cortex™ -A9 basé sur MPCore™ et une logique programmable (PL) Xilinx dans un seul appareil, construit sur une technologie de pointe haute la technologie des procédés haute performance et basse consommation. [20]

AXI BRAM

Le contrôleur AXI BRAM est un cœur IP souple à utiliser. Le cœur est conçu en tant qu'IP esclave de point de terminaison AXI pour l'intégration avec les périphériques d'interconnexion AXI et les systèmes maîtres afin de communiquer avec la BRAM locale. Le noyau prend en charge les transactions simples et en rafales vers BRAM et est optimisé pour la performance [21].

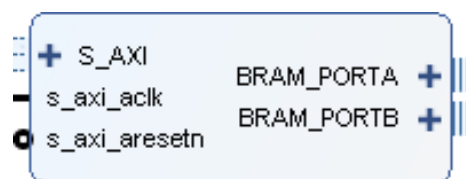


Figure 5.2 Ccontrôleurs AXI BRAM

Bus I2C

Le bus I2C (Inter Integrated Circuit Bus) est le bus historique, développé par Philips pour les applications de domotique et d'électronique domestique, notamment pour permettre de relier facilement à un microprocesseur avec les différents circuits intégrés [22].

Nous avons utilisé I2C vu qu'il est le plus épuré des bus de communication entre un processeur et ses périphériques, ainsi pour lire et programmer les capteurs externes.

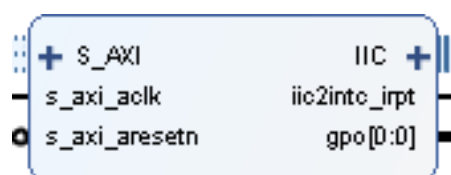


Figure 5.3 : Le Bus I2C

Port GPIO

Les ports GPIO (General Purpose Input/Output) soit entrée/sortie sont des ports qui sont très utilisés dans le monde des microcontrôleurs, en particulier dans le domaine de l'électronique embarquée. Selon la configuration, ces ports peuvent fonctionner aussi bien en entrée qu'en sortie [23].

Dans notre conception nous avons configuré 2 GPIO qui jouent le rôle des sorties (Boot et Reset)

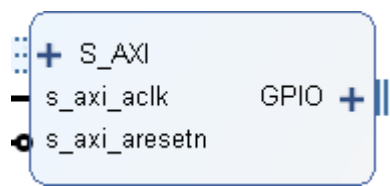


Figure 5.4 Le port GPIO

3. Composants du système

Capteur HTU21D

Le capteur HTU21D est un capteur numérique de température et d'humidité économique, facile à utiliser et extrêmement précis. Ce capteur est idéal pour la détection environnementale, l'enregistrement de données et est parfait pour les stations météorologiques ou les systèmes de contrôle d'humidification. Ce capteur permet l'obtention des relevés d'humidité relative et des relevés de température très précis [24].



Figure 5.5 Capteur HTU21D[25]

Microcontrôleur Minized

La carte MINIZED est une carte de développement avec une plate-forme de prototypage à faible coût. Cette conception compacte offre une connectivité embarquée via USB, Wi-Fi et Bluetooth. (Voir section 3.4).



Figure 5.6 Microcontrôleur Minized [26]

Carte de transmission/réception LL-RLP-20

Les modèles LL-RLP-20 est un module émetteur-récepteur RF bidirectionnels, optimisés pour une utilisation dans des bandes de fréquences spécifique. Il est intégré et idéal pour les réseaux étendus à faible consommation d'énergie (LPWAN) avec des exigences de bande passante réduite [27].

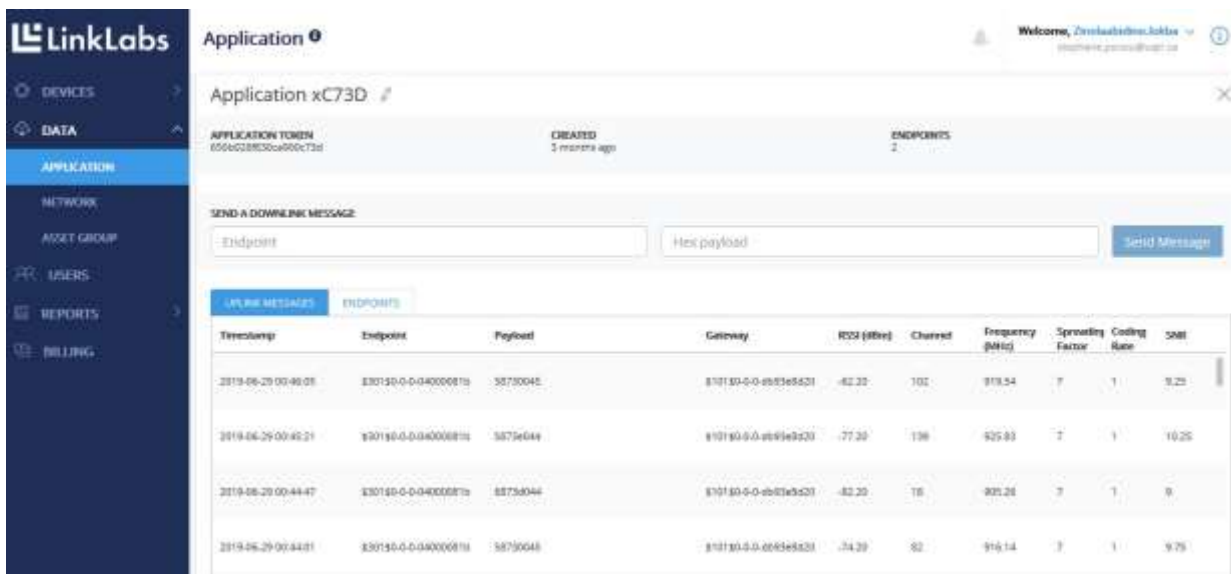


Figure 5.7 La carte Tx/Rx LL-RLP-20[28]

Cloud conductor

Conductor, est une plateforme de services de données en nuage. Il expose les interfaces de programme d'application (API), qui offrent aux titulaires de compte une interface simple pour accéder à leurs réseaux, ces API facilitent la création d'applications Web pour surveiller et contrôler les réseaux machine à machine (M2M) et Internet des Objets (IoT).

Le système Conductor est actuellement déployé dans Amazon Web Services. la figure suivante illustre l'interface du cloud conductor [29].



The screenshot displays the LinkLabs Cloud Conductor web interface. On the left is a dark blue navigation sidebar with options: DEVICES, DATA, APPLICATION (selected), NETWORK, ASSET GROUP, USERS, REPORTS, and BILLING. The main content area shows details for an application named 'Application xC73D'. It includes an 'APPLICATION TOKEN' (E064C38E3D0ca996c73d), a 'CREATED' timestamp (3 months ago), and 'ENDPOINTS' (2). Below this is a 'SEND A DOWNLINE MESSAGE' section with input fields for 'Endpoint' and 'Hex payload', and a 'Send Message' button. At the bottom, there is a table with two tabs: 'LIFELINE MESSAGES' and 'ENDPOINTS'. The 'ENDPOINTS' tab is active, showing a table with columns: Timestamp, Endpoint, Payload, Gateway, RSSI (dBm), Channel, Frequency (MHz), Spreading Factor, Coding Rate, and SNR.

Timestamp	Endpoint	Payload	Gateway	RSSI (dBm)	Channel	Frequency (MHz)	Spreading Factor	Coding Rate	SNR
2019-06-29 00:40:01	830180-0-0-04000087b	58750045	810180-0-0-0803e8520	-62.20	102	919.54	7	1	9.25
2019-06-29 00:40:21	830180-0-0-04000087b	5875e044	810180-0-0-0803e8520	-77.20	136	925.93	7	1	10.25
2019-06-29 00:44:47	830180-0-0-04000087b	8875e044	810180-0-0-0803e8520	-82.20	10	901.26	7	1	9
2019-06-29 00:44:01	830180-0-0-04000087b	58750045	810180-0-0-0803e8520	-74.20	82	916.14	7	1	9.75

Figure 5.8 L'interface du cloud conductor

Passerelle LinkLabs

Il s'agit d'une passerelle IoT / M2M de qualité industrielle à utiliser avec les réseaux LoRa. Il est capable de prendre en charge des milliers de terminaux avec 8 canaux de réception simultanés [30].



Figure 5.9 Passerelle LinkLabs[31]

3.1. Principe de raccordement

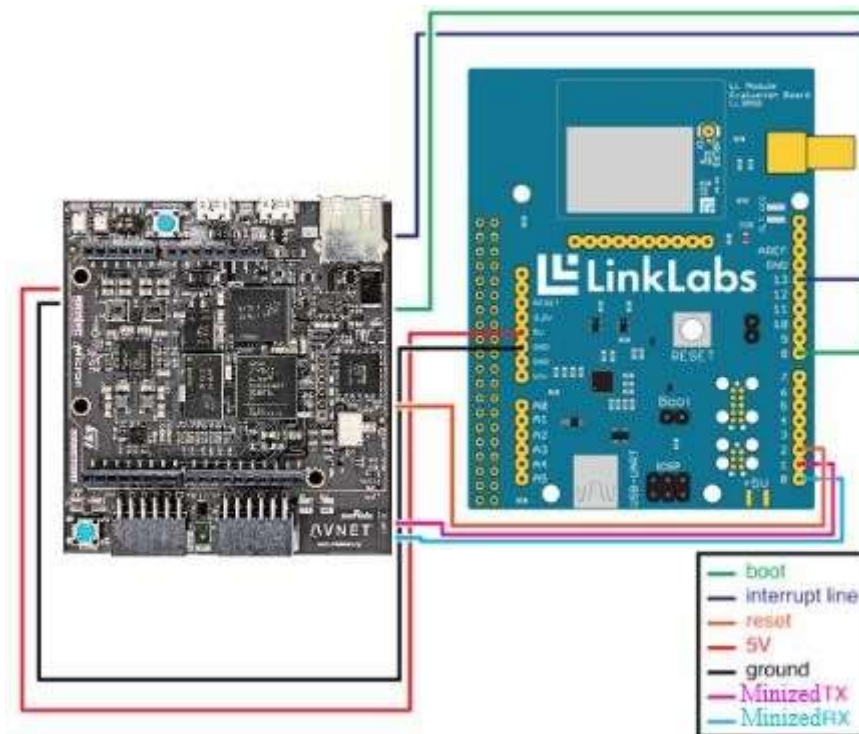


Figure 5.10 Schéma de raccordement LL-RLP-20 et Minized

Dans notre projet nous avons opté pour la carte microcontrôleur Minized. Malheureusement celle-ci ne permet pas la transmission des données via le protocole LoRaWAN. Pour résoudre ce petit problème on a fait appel à une carte de transmission/réception, afin de transmettre ces données vers le cloud. Nous avons choisi la carte LL-RLP-20 qui peut être configuré pour émettre / recevoir des signaux radiofréquences via son antenne de trace intégrée. La figure 5.5 illustre le schéma de raccordement.

4. Implémentation du Module de contrôle et de commande

4.1. Environnement d'exécution

Notre application est testée sur un ordinateur ayant les caractéristiques suivantes :

Configuration de la machine	Processeur	Intel(R) Core(TM) i5-8500T CPU@2.10GHZ 2.11GHZ
	Mémoire RAM	8.00 Go
	Nom du pc	HP ProDesk
Système d'exploitation	Windows 10 Enterprise © Microsoft Corporation, 2017 Edition 64bit	

Tableau 5.1 Caractéristiques du pc utilisé pour le projet

4.2. Outils et langage de développement

4.2.1. Outils de développement

Vivado

L'environnement de conception de la carte Minized, qui s'appelle Vivado Design Suite, est une suite logicielle produite par Xilinx pour la synthèse et l'analyse de conceptions HDL (*hardware description language*), il comporte des fonctionnalités supplémentaires pour le développement de système sur puce et la synthèse de haut niveau. Il représente également une réécriture et une refonte complète du flux de conception [32].

Xilinx SDK

Nous avons utilisé Le kit de développement logiciel Xilinx (XSDK) comme environnement de programmation, il constitue l'environnement de conception intégré permettant de créer des applications embarquées sur les microprocesseurs primés de Xilinx. Le SDK est la première application IDE à fournir une analyse, un débogage et des performances multiprocesseurs homogènes et hétérogènes [33].

4.2.2. Langage de développement

Langage C

Le **C** est un langage de programmation de bas niveau très populaire, créé dans les années 1970 par D.Ritchie et B.W.Kernighan. Il est portable, libre et rapide. Le **C** n'est sans doute pas le langage le plus facile à apprendre, ni le plus récent, mais ses qualités font de lui un langage incontournable en matière de programmation. [34]

Matlab

Matlab est un langage de développement informatique particulièrement dédié aux applications scientifiques. Il est utilisé pour développer des solutions nécessitant une très grande puissance de calcul. Nous avons utilisé **Matlab** pour appliquer les algorithmes d'apprentissage automatique et d'apprentissage profond sur les données.[35]

Python

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. **Python** est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. [36] Dans notre travail un script python est utilisé pour extraire les données à partir du cloud.

5. Résultats et discussion

Pour former des modèles de localisation, cinq techniques d'apprentissage automatique sont utilisées et comparées afin d'extraire le meilleur parmi eux. Les modèles sont formés pour les machines à vecteurs de support (SVM), les arbres de décision (DT), les classificateurs de forêt aléatoire (RF), le voisin K le plus proche (KNN) et les réseaux de neurones artificiels (ANN) Ces modèles sont implémentés dans Matlab.

Ces algorithmes nécessitent des données de formation et d'évaluation, indépendamment de leur champ d'application et de leur méthode. Après avoir obtenu ces données, les traiter et appliquer ces algorithmes sur eux, c'est le temps d'analyser et évaluer la performance de ces algorithmes. Pour cette raison là nous utilisons la matrice de confusion qui nous résume les

résultats de prédictions sur le problème de classification afin de comprendre de quelle façon ces modèles de classification sont confus lorsqu'ils effectuent des prédictions.

Dans ce qui suit les résultats obtenu après avoir appliqués ces algorithmes seulement sur les données de signaux puis sur les données de signaux et environnementaux.

5.1. Résultats de l'apprentissage automatique

5.1.1. Résultats d'application sur les données de signaux

Chaque ligne dans ces matrices correspond à une class réelle et chaque colonne correspond à une classe estimée. Par exemple dans la matrice de KNN la classe « C » est estimé avec 69% avec un taux d'erreur de 1% pour la classe « A » et de 31% pour la classe « B ».

La deuxième classe qui est la classe « A » est estimé avec 68% avec un taux d'erreur de 1% pour la classe « C » et 31 % pour la classe B.

La dernière classe « B » est estimé avec 61%, avec un taux d'erreur de 15% pour la classe « C », et 24% pour la classe « A ».

La même interprétation est appliqué aux autres algorithmes (matrices).la couleur blanche indique que l'algorithme n'admet pas une erreur d'estimation pour cette classe 0%, ce qui montre qu'il est plus performant que les autres.

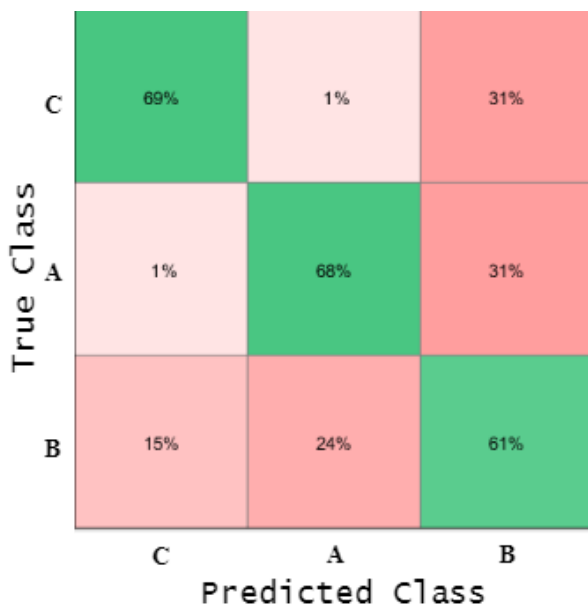


Figure 5.11 Matrice de confusion de KNN

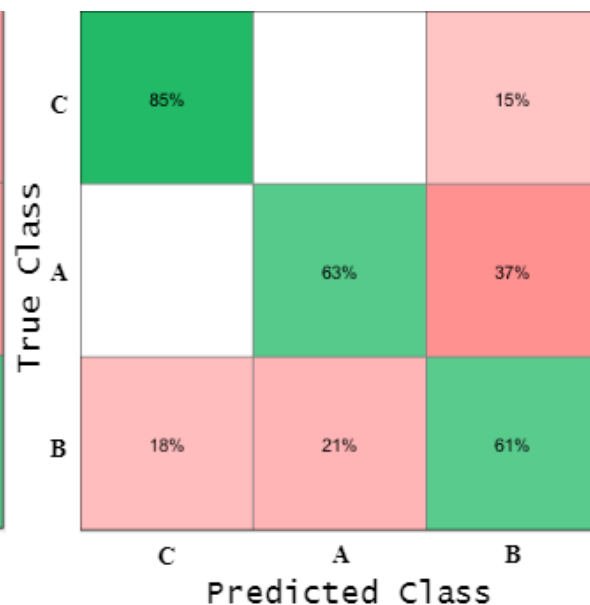


Figure 5.12 Matrice de confusion de SVM

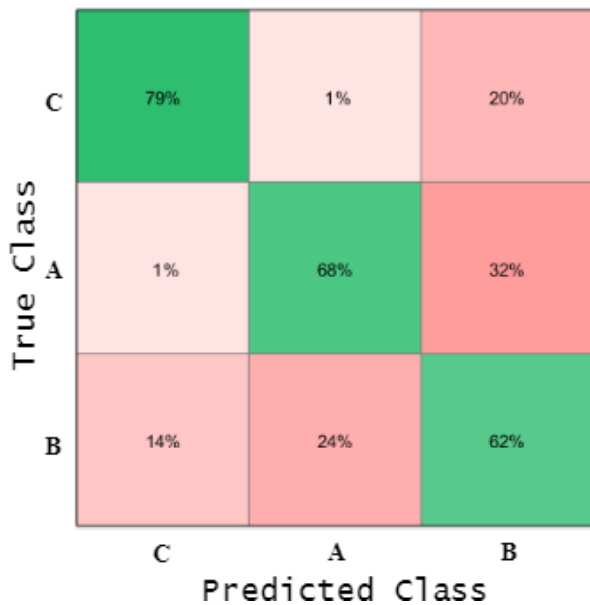


Figure 5.13 Matrice de confusion de DT

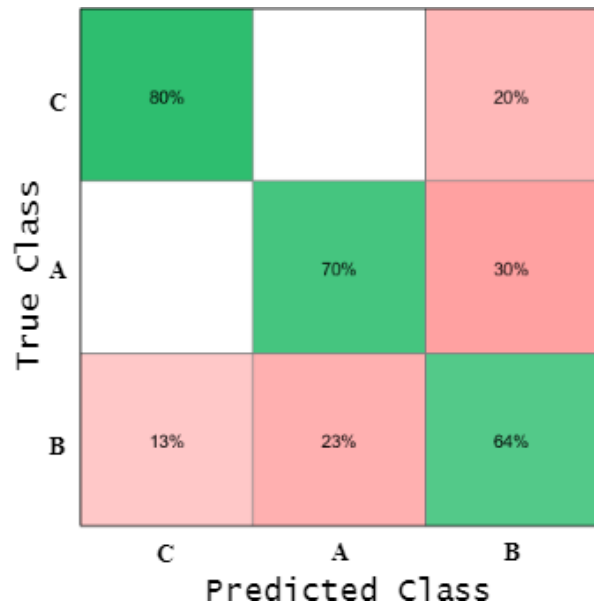


Figure 5.14 Matrice de confusion des RF

Le tableau ci-dessous montre la précision de la prédiction obtenue par les modèles précédents allant de faible à élever ainsi l'erreur de prédiction allant dans le sens inverse :

Ou :

$$\text{Précision totale} = \frac{\sum \text{T.V}}{\sum \text{F.V} + \text{T.V}}$$

Avec :

T.V : les bonnes valeurs de prédiction.

F.V : les erreurs de prédiction.

	KNN	SVM	DT	RF
Précisions (Accuracy %)	65.78%	69.43%	69.66%	71.33%
Erreur de prédiction	34.22%	30.57%	30.34%	28.67%

Tableau 5.2 Résultats d'application des algorithmes d'apprentissage sur les données de signaux.

La forêt aléatoire atteint la précision la plus élevée (71.33%) sur l'ensemble de données, tandis que les classificateurs DT et SVM se situent autour de 69%. Cependant, le classificateur KNN atteint le faible des précisions pour ces données (65.78%).

5.1.2. Résultats d'application sur les données de signaux et environnementales

Dans cette section on fait de même que la section précédente sauf qu'une donnée environnementale (Température) est ajoutée aux données de signaux précédents.

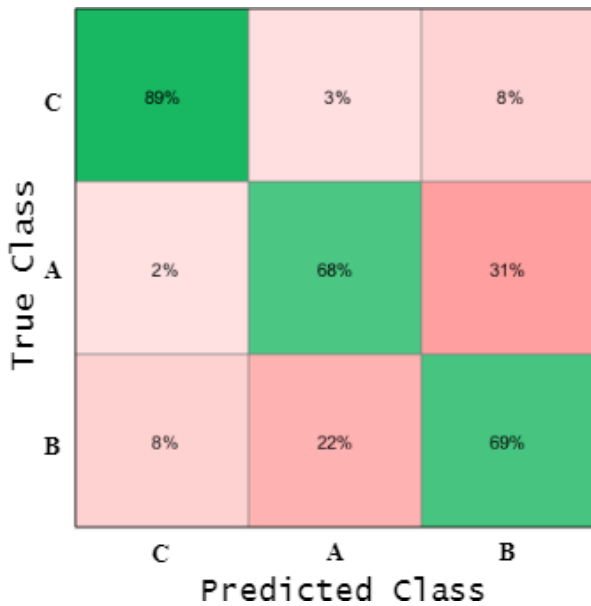


Figure 5.15 Matrice de confusion de KNN

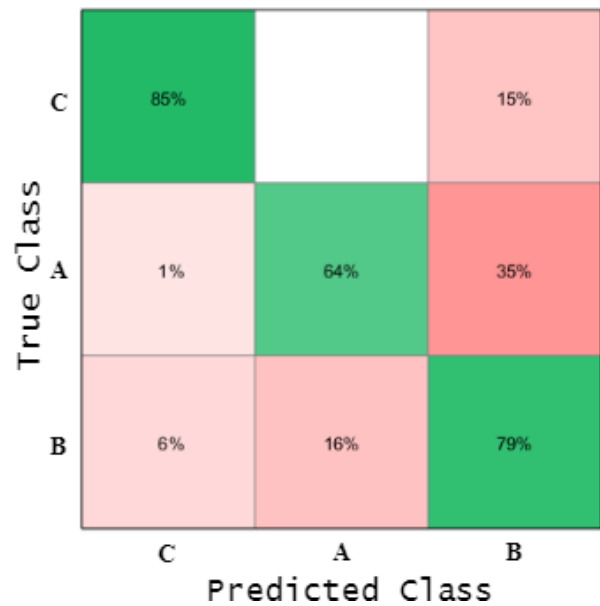


Figure 5.16 Matrice de confusion de SVM

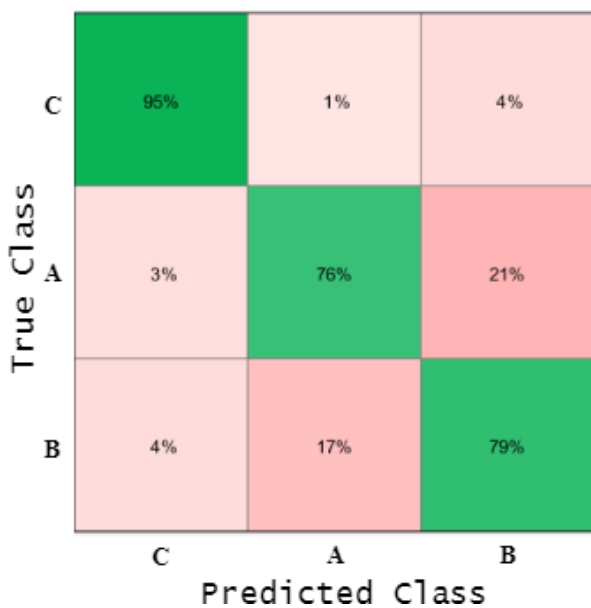


Figure 5.17 Matrice de confusion de DT

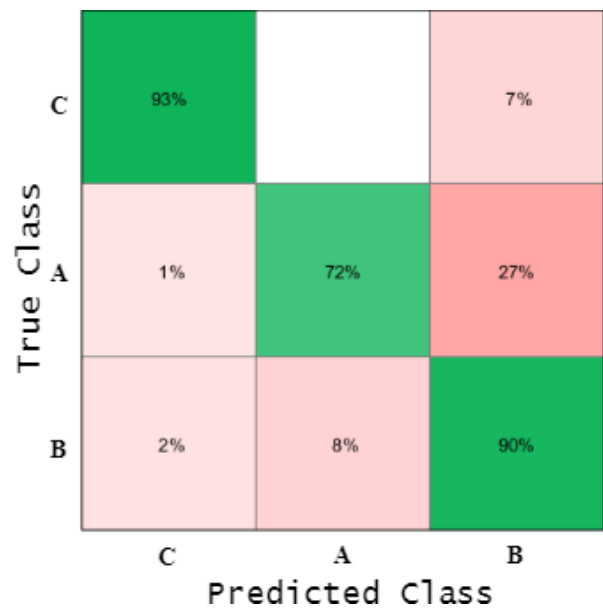


Figure 5.18 Matrice de confusion de RF

Le tableau 5.3 montre la précision de la prédiction obtenue par les modèles précédents allant de faible à élever ainsi l'erreur de prédiction allant dans le sens inverse :

	KNN	SVM	DT	RF
Précisions (Accuracy %)	75.33%	75.74%	83.33%	85%
Erreur de prédiction	24.67%	24.26%	16.67%	15%

Tableau 5.3 Résultats d'application des algorithmes d'apprentissage sur les données de signaux et environnementaux.

D'après la comparaison du tableau 5.2 et tableau 5.3. Nous remarquons que le taux de précision est augmenté d'une moyenne de 10.8% ce qui prouve que les données environnementaux ont un facteur important dans le cadre de la prédiction de localisation en intérieur.

5.2. Résultats de l'apprentissage profond

Les techniques d'apprentissage automatique, telles que les réseaux de neurones profonds, sont devenues un outil important pour plusieurs applications telles que la classification. Ces techniques ont atteint une précision prédictive extrêmement élevée. La figure 5.19 illustre la précision avec un simple réseau de neurones artificiel qui est 90.08% avec un taux d'apprentissage égal à 10^{-3} , ainsi la fonction de perte (lossfunction) qui est négligeable et tend vers le zéro, ce qui montre la puissance des réseaux de neurones dans ces problèmes de classification.

Dans de réseau de neurones nous avons utilisé une seule couche caché qui comprend 80 nœuds interne, nous avons utilisé également la fonction softmax comme une fonction d'activation et Adam comme une fonction d'optimisation, tandis que les poids et les bais sont généré aléatoirement par le réseau.

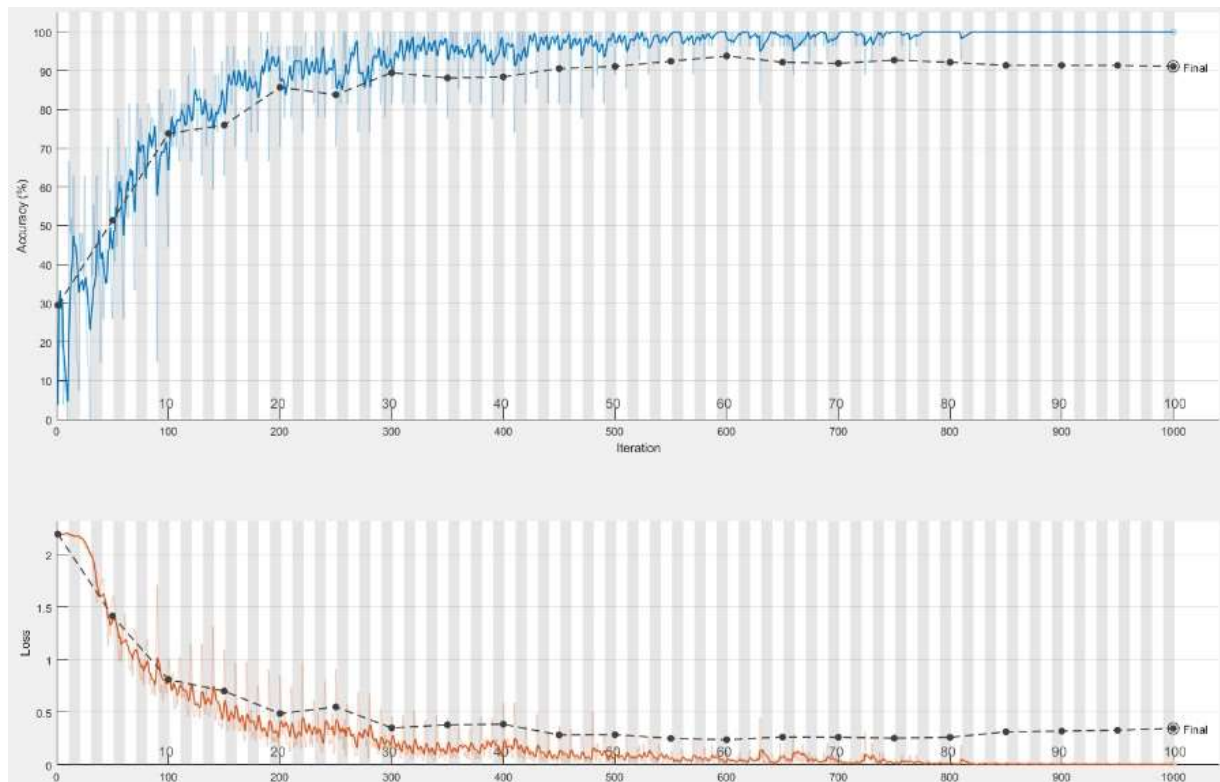


Figure 5.19 Résultats de précision et de fonction de perte d'un réseau de neurones artificiel

6. Conclusion

Dans ce chapitre nous avons montré l'implémentation des technologies IdO dans le cadre de la prédiction de la localisation intérieure. Ce chapitre a été consacré également à la présentation des outils et les technologies manipulées ainsi que les résultats des tests et de validation à travers les différentes expérimentations.

Conclusion Générale

La localisation en intérieure (*Indoor localization*) a été le centre d'intérêt et de recherche de plusieurs projets ces dernières années. D'énormes progrès ont été accomplis grâce aux avancées enregistrés dans plusieurs domaines comme ceux d'internet des objets, intelligence artificielle, d'informatique omniprésente, des systèmes ambiants, et aussi la miniaturisation des composants électronique. Malgré tout, il reste encore plusieurs défis à surmonter pour rendre possible l'implémentation de ce genre de problèmes en situation réelle et d'une façon très efficace.

Dans notre projet, nous avons mis en œuvre un prototype d'une plateforme IdO avec un système embarqué qui permet de répondre à quelque fonctionnalités que nous avons eu besoin à savoir l'acquisition de données par des capteurs qui se situent à l'intérieur d'un immeuble ainsi la transmission de ces données vers le cloud pour les stocké.

D'autre part, des algorithmes d'apprentissage automatique ont été utilisés dans le but de prédire la localisation en intérieure en se basant sur des données de signaux et environnementaux tel que l'indicateur de puissance du signal (RSSI), le rapport signal sur bruit (SNR), la fréquence (FQ), et la température.

Pour ce qui est de perspectives de continuation, plusieurs voies peuvent être envisagées. À titre indicatif, nous conseillons d'examiner et de développer les points suivants :

- Utiliser d'autres capteurs (d'autres données) qui aident à augmenter la précision de localisation.
- Profiter des avantages et des services du cloud avec l'exécution et le traitement des données à ce niveau.
- Faire plusieurs expérimentation dans plusieurs endroit (au lieu de 3 endroit).
- Utiliser une grande masse de donnée afin d'améliorer les résultats de test.
- Prendre en considération les personnes âgées et handicapés en mettant un système adaptatif selon l'emplacement de ses habitants

Bibliographie

ALGHURIED, Ahod. A model for anomalies detection in internet of things (IoT) using inverse weight clustering and decision tree. 2017.

ANI, R., KRISHNA, S., ANJU, N., et al. Iot based patient monitoring and diagnostic prediction tool using ensemble classifier. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017. p. 1588-1593.

ATZORI, Luigi, IERA, Antonio, et MORABITO, Giacomo. The internet of things: A survey. Computer networks, 2010, vol. 54, no 15, p. 2787-2805.

Arcep (2016). PRÉPARER LA RÉVOLUTION DE L'INTERNET DES OBJETS, Une cartographie des enjeux, 1–46 p

BENDIAB, Esma, KHOLLADI, Mohamed Khireddine. Vie artificielle, analyse, traitement et fouille de données. 2011. Thèse de doctorat, Université Mentouri, Constantine, Algérie.

BENGHOZI, Pierre-Jean, BUREAU, Sylvain, et MASSIT-FOLEA, Françoise. L'Internet des objets. Quels enjeux pour les Européens?, 2008.

CHENG, Yen-Kai, CHOU, Hsin-Jui, et CHANG, Ronald Y. Machine-learning indoor localization with access point selection and signal strength reconstruction. In : 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring). IEEE, 2016. p. 1-5.

Christophe Baland ,Damien Cauquil ,Thomas Gayet ,Julia Juvigny ,Renaud Lifchitz;Nha-Khanh Nguyen.(2016). Sécurité de l'Internet des Objets livre blanc.Paris:digital Security,44p

EVANS, Dave. The internet of things: How the next evolution of the internet is changing everything. CISCO white paper, 2011, vol. 1, no 2011, p. 1-11.

FATHY, Yasmin, BARNAGHI, Payam, et TAFAZOLLI, Rahim. Large-scale indexing, discovery, and ranking for the Internet of Things (IoT). ACM Computing Surveys (CSUR), 2018, vol. 51, no 2, p. 29.

FLACH, Peter. Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press, 2012.

GABRIELA, Moise, LOREDANA, Netedu, et ALINA, Toader Florentina. Bio-Inspired E-Learning Systems—A Simulation Case: English Language Teaching. In : Methodologies,

Tools and New Developments for E-Learning. IntechOpen, 2012.

GONG, Chunye, LIU, Jie, ZHANG, Qiang, et al. The characteristics of Cloud Computing. In : 2010 39th International Conference on Parallel Processing Workshops. IEEE, 2010. p. 275-279.

HODO, Elike, BELLEKENS, Xavier, HAMILTON, Andrew, et al. Threat analysis of IoT networks using artificial neural network intrusion detection system. In : 2016 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, 2016. p. 1-6.

Jiyi, PING, Lingdi, GE ITU, Series Y. global information infrastructure, internet protocol aspects and next-generation networks next generation networks–frameworks and functional architecture models. , 2001, p. 2060-201206WU,

MAHDAVINEJAD, Mohammad Saeid, REZVAN, Mohammadreza, BAREKATAIN, Mohammadamin, et al. Machine learning for Internet of Things data analysis: A survey. Digital Communications and Networks, 2018, vol. 4, no 3, p. 161-175.

MASOUDINEJAD, Mojtaba, VENKATAPATHY, Aswin Karthik Ramachandran, TONDORF, David, et al. Machine Learning Based Indoor Localisation Using Environmental Data in PhyNetLab Warehouse. In: Smart SysTech 2018; European Conference on Smart Objects, Systems and Technologies. VDE, 2018, p. 1-8

MELL, Peter, GRANCE, Tim, *et al.* The NIST definition of cloud computing. 2011. MIORANDI, Daniele, SICARI, Sabrina, DE PELLEGRINI, Francesco, et al. Internet of things: Vision, applications and research challenges. Ad hoc networks, 2012, vol. 10, no 7, p. 1497-1516.

QIAN, Huimin, MAO, Yaobin, XIANG, Wenbo, et al. Recognition of human activities using SVM multi-class classifier. Pattern Recognition Letters, 2010, vol. 31, no 2, p. 100-111.

ROTTENBERG, Sam. Modèles, méthodes et outils pour les systèmes répartis multiéchelles. 2015. Thèse de doctorat.

SALLOUHA, Hazem, CHIUMENTO, Alessandro, et POLLIN, Sofie. Localization in long-range ultra-narrow band IoT networks using RSSI. In: 2017 IEEE International Conference on Communications (ICC). IEEE, 2017. p. 1-6.

SHINDE, Tejaswinee A. ET PRASAD, Jayashree R. IoT based animal health monitoring with

naive Bayes classification. IJETT, 2017, vol. 1, no 2.

VERMESAN, Ovidiu, FRIESS, Peter, GUILLEMIN, Patrick, et al. Internet of things strategic research roadmap. Internet of things-global technological and societal trends, 2011, vol. 1, no 2011, p. 9-52.

WU, Tao et QIN, Kun. Inducing Uncertain Decision Tree via Cloud Model. In: 2009 Fifth International Conference on Semantics, Knowledge and Grid. IEEE, 2009. p. 85-91.

XIAO, Perry. Designing Embedded Systems and the Internet of Things (IoT) with the ARM mbed. John Wiley & Sons, Incorporated, 2018 p 3,4,23,24

Xiaoping, et al. Cloud storage as the infrastructure of Cloud Computing. In: 2010 International Conference on Intelligent Computing and Cognitive Informatics. IEEE, 2010. p. 380-383.

ZHANG, Qi, CHENG, Lu, et BOUTABA, Raouf. Cloud Computing: state-of-the-art and research challenges. Journal of internet services and applications, 2010, vol. 1, no 1, p. 7-18.

Webographie:

[1]. <https://www.silicon.fr/hub/hpe-intel-hub/les-differentes-couches-dune-infrastructure-iot>

[2]. https://en.wikipedia.org/wiki/Utility_computing

[3]. <http://www.barkatconsulting.com/consolidation-and-virtualization/>

[4]. <https://medium.com/@riameliaa25/klasifikasi-menggunakan-k-nearest-neighbor-knn-27a3db031248>

[5]. <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>

[6]. <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

[7]. https://www.einfochips.com/blog/hardware-design-challenges-of-the-embedded_internet-of-things-iot/

[8]. <https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>

[9]. https://ecoinfo.cnrs.fr/wp-content/uploads/2017/01/temperature_ecoinfo_20170316.pdf

[10]. https://fr.wikipedia.org/wiki/Banana_Pi.

-
- [11]. <http://www.siine.eu/decouverte-de-lorange-pi-pc/>.
- [12]. <https://fr.farnell.com/avnet/aes-minized-7z007-g/carte-de-dev-simple-coeur-zynq/dp/2775207>.
- [13]. <http://www.minized.org/>.
- [14]. <http://linuxgizmos.com/linux-driven-zynq-sbc-launches-at-89-with-wifi-bt-and-arduino/>.
- [15]. <https://whatis.techtarget.com/definition/sensor> consulté le.
- [16]. <https://www.futureelectronics.com/en/Microcontrollers/microcontrollers.aspx>.
- [17]. <https://www.techopedia.com/definition/32198/internet-of-things-gateway-iot-gateway>.
- [18]. <https://www.webopedia.com/TERM/T/transceiver.html>.
- [19]. <https://fr.audifanzine.com/techniques-du-son/forums/t.662434,ces-appareils-audio-etonnants,p.11.html>
- [20]. <http://www.knowres.ch/overview-of-products/#krm-3z7-module-family>
- [21]. https://www.xilinx.com/products/intellectualproperty/axi_bram_if_ctrl.html?fbclid=IwAR04ZjxZ4uLX3Dgwx87uAzmbZ6jpN4qpNDaXiKde6IEuqq4lWe1KMxuSvD .
- [22]. http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-Bus-I2C.htm?fbclid=IwAR3xxx1bB4pq4fI9casAw5Tk5GsB6MiK2E_w6lYuup4wMW8Tx0MzzQ8x_8
- [23]. https://www.windtopik.fr/debuter-gpio/?fbclid=IwAR3FT1S73XqNr1uzJv227GzADKpaow_8qC4pUF2-BSMh96cVbEnOzdztDA
- [24]. <http://www.baudaelectronica.com.br/sensor-de-temperatura-e-umidade-htu21d.html>
- [25]. <https://www.digikey.com/catalog/en/partgroup/htu21d-evaluation-boards/60497>
- [26]. <http://thinkhealthskincare.com/what-you-can-do-with-older-electronic-components/>
- [27]. <http://www.objectvalley.com/en/iotmodules.html>
- [28]. <https://www.digikey.com/product-detail/en/link-labs-inc/LL-RLP-20-USB-A/1616-1005-ND/5762385>.

- [29]. <https://www.link-labs.com/documentation/conductor-data-platform-user-guide>
- [30]. <https://www.link-labs.com/ll-bst-8-lora/symphony-gateway>
- [31]. <https://www.digikey.com/en/product-highlight/1/link-labs/symphony-link-8-channel-gateway>
- [32]. https://fr.qwerty.wiki/wiki/Xilinx_Vivado
- [33]. <https://www.xilinx.com/products/design-tools/embedded-software/sdk.html>
- [34]. http://www.depannetonpc.net/lexique/lire_70_langage-c.html?fbclid=IwAR1BjNx_AYNv0e5Yw2vUqSCgVPOyV-RamxGSR4ZL_Qnzl0CD0CAcxfiSZKg
- [35]. <https://www.jobintree.com/dictionnaire/definition-matlab-915.html?fbclid=IwAR1w1ONaoREuRbgYKYAhQOcQBiZNPynjld4Su9EzGE5dZrF5jBNETjslIo>
- [36]. https://docs.python.org/fr/3/tutorial/index.html?fbclid=IwAR0pSGsKGdYJGaKic3sSxA FE7Uea0GrRZsrtLeu8zj4T3ncKVU_YpQtspvU