

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique

CDTA



## Mémoire de Master

Spécialité : Réseaux et Télécommunications

Réalisé au sein du

Centre de Développement des Technologies Avancées (CDTA)

Réf : RT19

# Conception et évaluation d'un système de localisation intérieure utilisant LoRA

Réalisé par :

**Fekir Nawel**

&

**Kara Ahmed Fella**

Promoteurs :

- Dr DIB Wassila (CDTA)
- Dr KERDJIDJ Oussama(CDTA)

Co-Promotrice :

- Dr BERKAT Zeyneb (USDB-1)

Année Universitaire : 2024/2025

## REMERCIEMENTS

*Nous tenons tout d'abord à exprimer notre profonde gratitude envers Dieu, le Tout-Puissant et Miséricordieux, qui nous a donné la force, le courage, la volonté et la santé nécessaires pour mener à bien ce travail.*

*Nous souhaitons adresser nos sincères remerciements à notre promotrice, **Mme DIB Wassila**, pour avoir accepté de nous encadrer et pour sa disponibilité tout au long de la réalisation de ce mémoire. Ses précieux conseils, son expertise et sa bienveillance ont grandement contribué à la réussite de ce travail.*

*Nos remerciements vont également à **M. KERDJIDJ Oussama**, qui nous a accompagnés et soutenus tout au long de la réalisation de ce projet, aux côtés de Mme DIB, en sa qualité d'encadrant académique. Son aide, ses conseils techniques et sa disponibilité ont été d'un grand apport, et nous lui en sommes très reconnaissants.*

*Nous exprimons également notre profonde reconnaissance à notre Co-promotrice, **Mme BERKAT Zeyneb**, pour son accompagnement tout au long de ce travail. Sa disponibilité, ses conseils pertinents et son soutien constant ont grandement contribué à l'avancement et à la qualité de ce projet.*

*Nous remercions également le Centre de Développement des Technologies Avancées (CDTA) pour nous avoir offert l'opportunité de réaliser ce stage dans un environnement de travail favorable.*

*Nous tenons aussi à exprimer notre gratitude à l'ensemble de nos professeurs de l'**Université SAAD DAHLEB de BLIDA**, qui ont assuré notre formation tout au long de ces cinq années d'études. Leur dévouement, leur savoir-faire et leur accompagnement ont été essentiels dans notre parcours académique et professionnel.*

*Enfin, nous tenons à remercier toutes les personnes qui, de près ou de loin, ont contribué à l'élaboration de ce mémoire. Votre soutien, votre encouragement et votre collaboration ont été essentiels, et nous vous en sommes profondément reconnaissants.*

# **Dédicaces**

*Je dédie ce travail :*

*À mes chers parents, pour leur amour inconditionnel, leur soutien moral, leurs prières et leurs sacrifices, sans lesquels rien n'aurait été possible.*

*À mes sœurs Ikram, Asma et Zineb, mon frère Islam et ma belle-sœur Lamia pour leur patience, leurs encouragements et leur soutien psychologique qui ont été essentiels tout au long de ce parcours.*

*À mes chères nièces Ayoub, Yacine, Marie, Nour et Ines, sources de joie et de motivation dans les moments difficiles.*

*À mes cousines, HADJ RAHMOUN Oum El Kheir et BENTLEMCEN Maouia, chercheuses au CDTA, pour leurs conseils bienveillants, leur motivation et leurs orientations précieuses lors de mon stage.*

*À mes amis et collègues, pour leur aide, leurs conseils, et les moments partagés pendant ces années d'études.*

*A ma binôme Fella et toute sa famille.*

*À celles et ceux qui ont joué un rôle, petit ou grand, dans la réussite de ce mémoire.*

*Merci à tous pour votre soutien et votre confiance.*

**Nawel**

# **Dédicaces**

*Un remerciement du fond du cœur à mes parents source de tendresse, de noblesse et d'affection, AICHA et ALI pour leur amour inconditionnel, leur soutien moral et leurs sacrifices.*

*A mes quatre chers frères, AMINE, ABDELLATIF, ISHAK et YUCEF, ainsi qu'à ma belle-sœur BESSMA et ma petite nièce INES, je vous remercie pour leur encouragement et leur présence constante à mes côtés.*

*Une dédicace spéciale à ma tante bien-aimée DJAMILA et à mon grand-père Mohammed, que Dieu leur fasse miséricorde, pour leurs encouragements et leurs prières tout au long de ma carrière universitaire.*

*Un merci particulier à mon ami REDHA pour son soutien moral et ses encouragements tout au long de cette période.*

*Enfin, et le meilleur pour la fin, une dédicace toute spéciale à mon binôme et amie, NAWEL. Pour son engagement, sa persévérance et son esprit d'équipe tout au long de ce travail.*

*Merci pour les heures passées ensemble à réfléchir, à corriger, à parfois douter mais surtout à avancer.*

*ET à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.*

*Merci du fond du cœur.*

**Fella**

## المخلص

يركز هذا البحث على استخدام تقنية LoRa لتحديد المواقع داخل المباني، خاصة في الحالات التي تكون فيها الأنظمة التقليدية مثل GPS غير فعالة. تعتمد الطريقة المقترحة على استخدام قراءات RSSI من عدة بوابات LoRa، مع خوارزميات تعلم آلي خاضعة للإشراف لتقدير موقع جهاز الإرسال. تم اختبار النظام في ثلاث بيئات مختلفة: مكتب، ممر، ومبنى متعدد الطوابق. تم تطوير نظام متكامل مادي وبرمجي، يشمل وحدات LoRa، ونظام جمع بيانات آلي باستخدام Raspberry Pi، وقاعدة بيانات مركزية. مكّنت البيانات المجمعة من تدريب وتقييم عدة نماذج تصنيف مثل KNN و SVM و Random Forest. أظهرت النتائج أن دقة التمرکز تتأثر بالبيئة الفيزيائية، وتموضع البوابات، واستراتيجية التصنيف. وقد حققت خوارزمية Random Forest أفضل أداء في جميع السيناريوهات. يبرز هذا العمل إمكانية استخدام LoRa مع التعلم الآلي لبناء أنظمة تموضع داخلية فعالة، وقابلة للتطوير، ومنخفضة الاستهلاك، مع فتح آفاق لتحسينها باستخدام تقنيات متقدمة مثل التعلم العميق أو التوضع التعاوني.

**الكلمات المفتاحية:** LoRa، RSSI، Raspberry Pi، التعلم الآلي، تحديد المواقع داخل المباني.

## RESUME

Ce mémoire s'intéresse à l'utilisation de la technologie LoRa pour la localisation en environnement intérieur, dans des contextes où les solutions classiques comme le GPS sont inadaptées. L'approche repose sur l'exploitation de mesures RSSI issues de plusieurs passerelles LoRa, combinées à des algorithmes d'apprentissage supervisé pour estimer la position d'un émetteur. Trois types d'environnements ont été étudiés : bureau, couloir et bâtiment à plusieurs étages. Une architecture complète, matérielle et logicielle, a été mise en place, intégrant des modules LoRa, un système de collecte automatisée via Raspberry Pi, et une base de données centralisée. Les données recueillies ont permis d'entraîner et d'évaluer plusieurs modèles de classification (KNN, SVM, Random Forest). Les résultats montrent que la précision de localisation dépend de l'environnement, de la disposition des passerelles et de la stratégie de classification utilisée. L'algorithme Random Forest a obtenu les meilleures performances dans l'ensemble des cas testés. Ce travail démontre le potentiel du couplage LoRa – machine learning pour concevoir des systèmes de localisation intérieure efficaces, économiques et évolutifs, tout en ouvrant la voie à de futures améliorations par des techniques avancées comme le Deep learning ou la localisation collaborative.

**Mots-clés :** LoRa, RSSI, apprentissage automatique, localisation en intérieur, Raspberry Pi

## ABSTRACT

This dissertation focuses on the use of LoRa technology for indoor localization, particularly in environments where conventional solutions like GPS are ineffective. The proposed approach combines RSSI measurements from multiple LoRa gateways with supervised machine learning algorithms to estimate the position of a transmitter. Three environments were considered: office, hallway, and multi-story building. A complete hardware and software system was developed, integrating LoRa modules, an automated data collection setup using Raspberry Pi, and a centralized database. The collected data enabled the training and evaluation of various classification models (KNN, SVM, Random Forest). Results indicate that localization accuracy is highly influenced by the physical environment, gateway configuration, and the chosen classification strategy. Random Forest consistently achieved the highest performance across all scenarios. This work highlights the potential of integrating LoRa with machine learning to develop efficient, scalable, and low-power indoor positioning systems, and paves the way for future enhancements using deep learning or collaborative localization.

**Keywords:** LoRa, RSSI, machine learning, indoor positioning, Raspberry Pi.

# Table des matières

LISTE DES FIGURES.....	viii
LISTE DES TABLEAUX.....	ix
LISTE DES ABREVIATIONS.....	x
INTRODUCTION GENERALE .....	1
CHAPITRE 1 : LES SYSTEMES DE LOCALISATION EN INTERIEUR.....	3
1.1 Introduction .....	4
1.2 Concepts Fondamentaux De La Localisation En Intérieur.....	4
1.2.1 Technologies de localisation.....	6
1.2.2 Grandeurs physique utilisées pour la localisation .....	8
1.2.3 Algorithmes de localisation.....	9
1.3 Systèmes de localisation avec LoRa.....	10
1.4 Conclusion.....	12
CHAPITRE 2 : CADRE TECHNOLOGIQUE ET ALGORITHMIQUE POUR LA LOCALISATION EN INTERIEUR PAR RSSI ET LORA.....	13
2.1 Introduction .....	14
2.2 Technologie Lora/LoRaWAN.....	14
2.3 L'indicateur De Puissance Du Signal Reçu (RSSI).....	17
2.4 L'apprentissage Automatique (Machine Learning).....	18
2.4.1 Algorithmes sélectionnés .....	19
2.5 Conclusion.....	25
CHAPITRE 3 : IMPLEMENTATION MATERIELLE ET LOGICIEL.....	26
3.1 Introduction .....	27
3.2 Schéma Synoptique Du Système :.....	27
3.2.1 Architecture Matériel de la solution.....	28
3.3 Traitement De Données.....	38
3.4 Conclusion.....	40
CHAPITRE 4 : MESURES, TRAITEMENT ET RESULTATS.....	41
4.1 Introduction .....	42

4.2	Compagnes De Mesure.....	42
4.3	Acquisition Des Données .....	44
4.4	Évaluation Des Performances.....	45
4.4.1	Scénario 1.....	47
4.4.2	Scénario 2.....	51
4.4.3	Scénario 3.....	53
4.5	Discussion Des Résultats.....	55
4.6	Conclusion.....	57
	CONCLUSION GENERALE.....	58
	BIBLIOGRAPHIES .....	59

## LISTE DES FIGURES

<b>Figure 1.1</b> : Couverture des technologies de localisation interieur. ....	5
<b>Figure 1.2</b> : Synthèse des approches en localisation en intérieur. ....	6
<b>Figure 1.3</b> : Différents technologies de localisation. ....	6
<b>Figure 1.4</b> : Environnement intérieure (zone fermée). ....	10
<b>Figure 2.1</b> : LoRa et LoRaWAN. ....	14
<b>Figure 2.2</b> : Architecture de LoRaWan. ....	15
<b>Figure 2.3</b> : Visualisation des chirps dans la modulation LoRa. ....	16
<b>Figure 2.4</b> : Types d'apprentissage automatique. ....	19
<b>Figure 2.5</b> : Processus de Random forest. ....	23
<b>Figure 3.1</b> : Schéma synoptique du système. ....	27
<b>Figure 3.2</b> : Schéma électrique du Nœud mobile. ....	28
<b>Figure 3.3</b> : Module ESP32. ....	28
<b>Figure 3.4</b> : Module LoRa. ....	29
<b>Figure 3.5</b> : Arduino IDE. ....	30
<b>Figure 3.6</b> : Schéma électrique de la passerelle. ....	31
<b>Figure 3.7</b> : Raspberry Pi4. ....	32
<b>Figure 3.8</b> : Message d'initialisation. ....	36
<b>Figure 3.9</b> : Organigramme de nœud mobile sous ESP32. ....	36
<b>Figure 3.10</b> : Organigramme de passerelle sous Raspberry Pi. ....	37
<b>Figure 4.1</b> : Environnement expérimental du Scénario 1. ....	42
<b>Figure 4.2</b> : Environnement expérimental du Scénario 2. ....	43
<b>Figure 4.3</b> : Environnement expérimental du Scénario 3. ....	44
<b>Figure 4.4</b> : Images des expérimentations. ....	44
<b>Figure 4.5</b> : Collecte des données. ....	45
<b>Figure 4.6</b> : Performances du scénario 1 pour différents tailles de segment. ....	47
<b>Figure 4.7</b> : Division en zone_S1. ....	48
<b>Figure 4.8</b> : Performances selon les stratégies 1 et 2_S1. ....	48
<b>Figure 4.9</b> : Courbes ROC des stratégies 1 et 2_S1. ....	49
<b>Figure 4.10</b> : Performance selon la stratégie 3_S1. ....	49
<b>Figure 4.11</b> : Courbes ROC de la stratégie 3_S1. ....	50
<b>Figure 4.12</b> : Division en zone_S2. ....	51
<b>Figure 4.13</b> : Performances selon les stratégies 1 et 2_S2. ....	52
<b>Figure 4.14</b> : Courbes ROC _S2. ....	52
<b>Figure 4.15</b> : Performances selon les stratégies 1 et 2_S3. ....	54
<b>Figure 4.16</b> : Courbes ROC _S3. ....	54

## LISTE DES TABLEAUX

<b>Tableau 3.1</b> : trame des données LoRA. ....	36
<b>Tableau 4.1</b> : Les caractéristiques temporelles. ....	46

## LISTE DES ABREVIATIONS

<b>Abréviation</b>	<b>Signification</b>
AoA	Angle of Arrival
AUC	Area Under Curve
BLE	Bluetooth Low Energy
CSS	Chirp Spread Spectrum
DBm	Decibel-milliwatts
FHSS	Frequency Hopping Spread Spectrum
FSPL	Free Space Path Loss
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
IoT	Internet of Things
IPS	Indoor Positioning System
ISM	Industrial, Scientific and Medical (frequency band)
KNN	K-Nearest Neighbors
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LOS	Line of Sight
LPWAN	Low Power Wide Area Network
MAC	Medium Access Control
ML	Machine Learning
NFC	Near Field Communication
RF	Random Forest
ROC	Receiver Operating Characteristic
RSSI	Received Signal Strength Indicator
SF	Spreading Factor

<b>Abréviation</b>	<b>Signification</b>
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
TDoA	Time Difference of Arrival
ToA	Time of Arrival
TX	Transmission
RX	Réception
Wi-Fi	Wireless Fidelity
W	Windowing

## INTRODUCTION GENERALE

La localisation en intérieur (Indoor Positioning System, IPS) représente aujourd'hui un enjeu majeur dans de nombreux domaines tels que la logistique, la domotique, la sécurité ou encore les services contextuels dans les bâtiments intelligents. Si le GPS s'est imposé comme une référence pour la localisation en extérieur, ses performances se dégradent fortement en milieu clos en raison de l'atténuation des signaux satellites.

Pour pallier ces limitations, plusieurs technologies ont été étudiées dans la mise en œuvre des systèmes de positionnement intérieur, telles que le Wi-Fi, le Bluetooth, le Zigbee, l'UWB ou encore le LoRa. Chacune présente des avantages et des contraintes en matière de coût, de consommation énergétique, de précision, de portée ou de complexité d'installation. Le choix de la technologie dépend ainsi étroitement des exigences spécifiques à chaque cas d'usage.

Dans ce contexte, la technologie **LoRa (Long Range)** se distingue par sa capacité à offrir des communications longue portée avec une consommation énergétique très faible, ce qui la rend particulièrement adaptée aux environnements vastes, multi-étages et aux dispositifs autonomes fonctionnant sur batterie. Toutefois, LoRa ne fournit pas nativement de mécanisme de localisation ; l'exploitation de paramètres tels que le **RSSI** ou le **SNR** reste complexe en raison de leur variabilité et de leur sensibilité aux interférences. Malgré ces contraintes, son faible coût d'infrastructure et sa bonne capacité de couverture en font une alternative prometteuse pour des solutions de localisation intérieure.

L'objectif principal de ce mémoire est d'étudier la faisabilité de l'utilisation de LoRa pour la localisation en intérieur, en combinant cette technologie avec des algorithmes d'apprentissage automatique afin d'estimer la position d'un émetteur à partir des mesures RSSI collectées. Pour cela, une architecture complète a été conçue, intégrant plusieurs passerelles LoRa et des nœuds mobiles. Des campagnes de mesures ont été menées dans trois environnements distincts : un bureau (Scénario 1), un couloir (Scénario 2) et un bâtiment à plusieurs étages (Scénario 3). Les données ont ensuite été utilisées pour entraîner et évaluer différents modèles supervisés, tels que K-Nearest Neighbors (KNN), Support Vector Machines (SVM) et Random Forest (RF), afin d'estimer la position de l'émetteur dans des configurations variées.

Pour structurer cette démarche, le mémoire est organisé en quatre chapitres complémentaires :

Le chapitre 1 présente les fondements des systèmes de localisation en intérieur, les technologies existantes et les principaux algorithmes utilisés.

Le chapitre 2 décrit le cadre technologique et algorithmique de l'étude, en détaillant la technologie LoRa, le rôle du RSSI, et les méthodes d'apprentissage automatique retenues.

Le chapitre 3 est consacré à l'architecture matérielle et logicielle du système proposé, avec une description du schéma global, des composants utilisés et de leur mise en œuvre.

Le chapitre 4 expose la méthodologie expérimentale, les résultats obtenus dans différents environnements intérieurs, ainsi qu'une analyse des performances des modèles de localisation.

En fin, une conclusion générale qui ouvre sur des perspectives d'amélioration et d'extensions possibles du système proposé.

**CHAPITRE -1 :**

**LES SYSTEMES DE LOCALISATION EN INTERIEUR**

## **1.1 Introduction**

Les systèmes de localisation en intérieur (Indoor Positioning Systems, IPS) constituent une réponse technologique aux limitations des systèmes globaux de navigation par satellite (GNSS), tels que le GPS, dans les environnements fermés. Bien que les technologies satellitaires soient très performantes pour la localisation en milieu extérieur, leur efficacité chute drastiquement en intérieur en raison de l'atténuation ou du blocage total des signaux par les structures bâties.

Dans ce contexte, plusieurs technologies alternatives exploitant des signaux radiofréquences à courte ou longue portée, telles que le Wi-Fi, le Bluetooth Low Energy (BLE), ou encore LoRa, ont été développées afin d'assurer une localisation fiable à l'intérieur des bâtiments. Ces systèmes sont utilisés dans des domaines variés, allant de la logistique à la santé, en passant par les services de géolocalisation contextuelle. Toutefois, cette diversité soulève des défis considérables en matière de précision, de coût de déploiement, de consommation énergétique et d'interopérabilité.

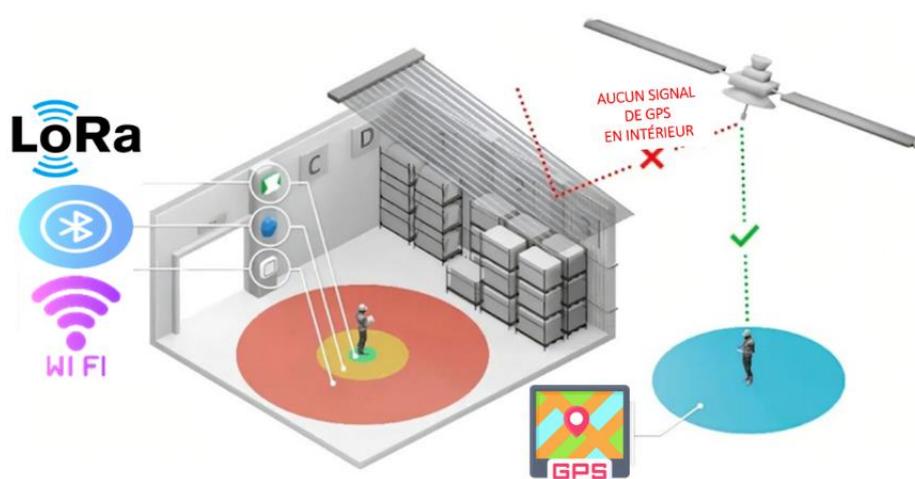
Ce chapitre vise à présenter un état de l'art des technologies de localisation en intérieur, les technologies sous-jacentes, et les performances associées. Il met également en lumière les avantages et limitations de chaque méthode, afin de mieux orienter les choix technologiques selon les besoins applicatifs. Une attention particulière sera ensuite accordée à la technologie LoRa, reconnue pour sa portée étendue et sa faible consommation énergétique, qui la rend pertinente pour des scénarios de localisation complexes et à grande échelle.

## **1.2 Concepts Fondamentaux De La Localisation En Intérieur**

La géolocalisation consiste à déterminer la position avec une certaine précision d'un objet ou d'une personne à l'aide de diverses technologies (Global Positioning system (GPS), réseaux cellulaires, BLE, LoRa, etc.). Aujourd'hui incontournable, elle impacte tous les aspects de notre vie quotidienne ; que ce soit pour des applications grand public tel le suivi de colis, navigation urbaine, services de proximité ou pour des usages professionnels tel l'optimisation logistique, gestion des stocks et sécurité des travailleurs [1].

Son rôle stratégique s'étend à des secteurs clés comme la santé (suivi médical), l'industrie (traçabilité) ou les services d'urgence (localisation en temps réel). Ces applications

démontrent comment la géolocalisation est devenue un outil essentiel pour la prise de décision et l'efficacité opérationnelle dans de nombreux domaines.



**Figure 0.1** : Couverture des technologies de localisation intérieur.

La géolocalisation peut être globalement divisée en deux grandes catégories selon le contexte d'utilisation, Comme illustré dans la figure 1.1 [2] :

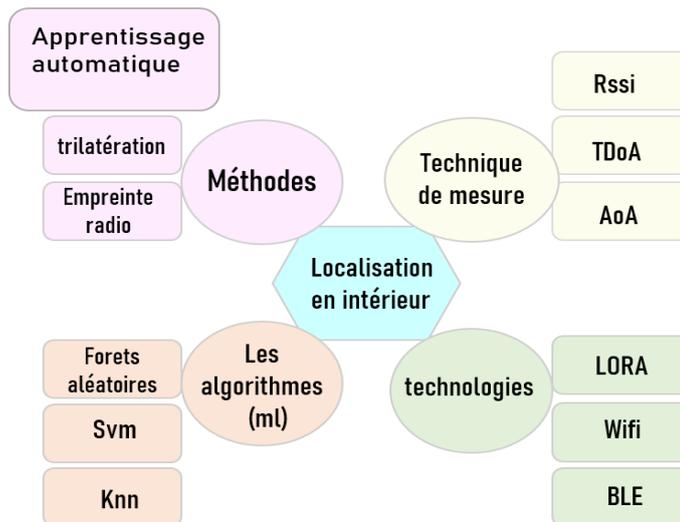
**a) Localisation en extérieur (Outdoor Localisation)**

La localisation extérieure repose principalement sur les signaux issus de constellations satellitaires telles que le GPS et GNSS (Global Navigation Satellite System), permettant une couverture globale avec une précision pouvant atteindre quelques mètres. Elle est particulièrement adaptée aux espaces ouverts et dégagés.

**b) Localisation en intérieur (Indoor Localisation)**

Dans les environnements clos (hôpitaux, entrepôts, centres commerciaux, etc.), les signaux satellitaires sont fortement atténués ou totalement bloqués. Pour pallier cette limitation, des solutions spécifiques dites IPS ont été développées. Celles-ci utilisent des signaux radiofréquence émis par des infrastructures locales, comme des points d'accès Wi-Fi, des balises BLE, ou encore des passerelles LoRa, afin d'estimer la position d'un terminal [3].

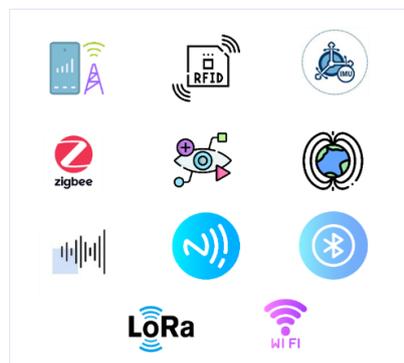
La figure ci-dessous 1.2 présente une vue d'ensemble des principales méthodes, techniques de mesure, technologies et algorithmes d'apprentissage automatique utilisés dans les systèmes de localisation en intérieur.



**Figure 0.2** : Synthèse des approches en localisation en intérieur.

### 1.2.1 Technologies de localisation

Les IPS reposent sur une grande diversité de technologies. Comme le montre la figure 1.3, chacune présentant des caractéristiques distinctes en termes de précision, de portée, de robustesse aux interférences et de coût d'implémentation. Le choix de la technologie dépend fortement du contexte d'utilisation, des contraintes environnementales, et des exigences applicatives (temps réel, précision, autonomie, etc.) [4].



**Figure 0.3** : Différents technologies de localisation.

Parmi les technologies utilisées, on retrouve :

- a. **Wi-Fi** : Largement déployé, le Wi-Fi permet une localisation avec une précision modérée (environ 5 à 15 mètres) en exploitant la puissance du signal reçu (RSSI). Son principal atout réside dans sa disponibilité, ce qui le rend attractif pour des systèmes à faible coût d'infrastructure [4].

- b. **ZigBee** : Technologie de communication à faible puissance, ZigBee offre une portée modeste (jusqu'à 100 m en champ libre) et est adaptée aux environnements où l'économie d'énergie est cruciale. Toutefois, sa précision de localisation reste limitée [5].
- c. **RFID (Radio-Frequency Identification)** : Principalement utilisée pour la traçabilité d'objets, la RFID permet l'identification sans contact via des tags passifs ou actifs. Bien que très efficace pour le suivi ponctuel, elle est moins adaptée au suivi en temps réel de mouvements continus [6].
- d. **BLE (Bluetooth Low Energy)** : Offrant un bon compromis entre précision (1 à 2 mètres) et consommation énergétique, BLE est très utilisé dans les environnements mobiles et portables. Il permet une localisation de proximité à faible coût [4].
- e. **UWB (Ultra-Wide Band)** : Cette technologie fournit une précision de localisation de l'ordre du centimètre grâce à sa large bande passante. UWB est particulièrement adaptée aux applications exigeant une grande précision, telles que la robotique ou la réalité augmentée, mais elle implique un coût plus élevé [4].
- f. **Radio à longue portée (LoRa, Sigfox)** : Ces technologies LPWAN permettent une couverture étendue avec une consommation énergétique minimale. Bien que leur précision soit inférieure à celle d'autres systèmes leur portée les rend utiles dans les scénarios nécessitant une couverture d'un bâtiment entier ou d'un campus [7].
- g. **NFC (Near Field Communication)** : Utilisée pour des échanges sécurisés à très courte distance (<10 cm), la NFC est idéale pour les applications nécessitant une identification immédiate et fiable, bien que non adaptée à un suivi de position [8].
- h. **Réseaux cellulaires (2G/3G/4G/5G)** : Ces réseaux offrent une large couverture et sont de plus en plus utilisés dans des approches hybrides de localisation. Toutefois, ils impliquent une consommation énergétique importante et une précision variable (quelques dizaines à centaines de mètres en intérieur) [9].
- i. **IMU (Inertial Measurement Units)** : Composées d'accéléromètres et de gyroscopes, les IMU permettent le suivi des mouvements en temps réel sans infrastructure. Elles sont sensibles à la dérive temporelle, ce qui nécessite souvent une recalibration ou une fusion de capteurs [4].

- j. Signaux acoustiques** : Utilisant des ultrasons ou des sons audibles, ces systèmes permettent une localisation précise sans recours à des ondes radio, ce qui peut être avantageux en matière de respect de la vie privée. Toutefois, ils sont sensibles aux interférences sonores et aux propriétés acoustiques des lieux [10].
- k. Champs magnétiques** : L'exploitation des anomalies du champ magnétique terrestre ou généré par les structures du bâtiment permet une localisation sans infrastructure additionnelle. Cette méthode, bien que prometteuse, est sujette à des variations locales difficilement modélisables [9].
- l. Méthodes visuelles (caméras, vision par ordinateur)** : Ces approches offrent une grande précision et peuvent fonctionner sans infrastructure réseau, mais elles nécessitent des ressources de calcul importantes et sont sensibles aux changements de l'environnement (lumière, objets mobiles) [11].

### 1.2.2 Grandeurs physique utilisées pour la localisation

Les performances des systèmes de localisation en environnement intérieur dépendent fortement des techniques de mesure utilisées pour estimer la position d'un dispositif. Parmi les approches les plus courantes, on retrouve le Received Signal Strength Indicator (RSSI), l'Angle d'Arrivée (AoA), le Temps d'Arrivée (ToA) et la Différence de Temps d'Arrivée (TDoA). Chacune de ces méthodes repose sur des caractéristiques physiques différentes du signal et présente des avantages ainsi que des limitations spécifiques [4].

- a) RSSI** : Le RSSI mesure la puissance du signal reçu et établit une relation empirique entre cette puissance et la distance entre l'émetteur et le récepteur. Cette technique est largement utilisée en raison de sa simplicité et de son faible coût d'implémentation, notamment avec les technologies Wi-Fi, BLE ou ZigBee. Toutefois, sa précision peut être fortement affectée par les obstacles, les réflexions, et les interférences multipaths, rendant parfois l'estimation de distance instable dans des environnements complexes [9].
- b) AoA** : L'angle d'arrivée est déterminé à partir de la direction sous laquelle le signal parvient à une antenne ou un réseau d'antennes. En utilisant plusieurs antennes espacées, il est possible d'estimer cet angle grâce à la différence de phase ou de temps de réception du signal. Combiné à la position connue du point d'accès, AoA permet une localisation précise par triangulation. Cette méthode offre une bonne précision angulaire

mais nécessite des équipements plus complexes, notamment des antennes directionnelles ou des réseaux MIMO [9].

- c) **ToA** : Cette technique repose sur la mesure absolue du temps mis par un signal pour parcourir la distance entre l'émetteur et le récepteur. En connaissant la vitesse de propagation du signal, on peut en déduire la distance. Pour obtenir une position, plusieurs points d'accès sont nécessaires. Le principal défi du ToA réside dans la nécessité d'une synchronisation temporelle très précise entre l'émetteur et le récepteur, ce qui peut être difficile à assurer dans des systèmes distribués [9].
- d) **TDoA** : Contrairement au ToA, la méthode TDoA mesure la différence de temps d'arrivée d'un même signal reçu par plusieurs récepteurs synchronisés. Cette différence permet de déterminer des hyperboles sur lesquelles le dispositif se trouve, et l'intersection de plusieurs hyperboles permet d'estimer sa position. Le TDoA est moins contraignant en matière de synchronisation que le ToA, mais nécessite tout de même une infrastructure de réception synchronisée [12].

### 1.2.3 Algorithmes de localisation

Les algorithmes de localisation constituent un élément clé dans l'estimation de la position d'un dispositif mobile à l'intérieur des bâtiments. Ils exploitent les données issues de diverses technologies de détection (Wi-Fi, BLE, UWB, IMU, etc.) pour déterminer la position spatiale d'un objet ou d'une personne. Ces méthodes peuvent être classées en plusieurs catégories principales : géométriques, basées sur des empreintes (fingerprinting), probabilistes, et fondées sur l'apprentissage automatique.

- a. **Trilatération** : Cette approche géométrique repose sur la mesure des distances entre le dispositif à localiser et plusieurs points d'accès (ou balises) dont les positions sont connues. En exploitant les intersections de sphères ou cercles formés à partir de ces distances, la position du dispositif est calculée. Cette méthode est simple mais peut être sensible au bruit des signaux et aux erreurs de synchronisation [13].
- b. **Empreinte digitale (Fingerprinting)** : Elle consiste à établir une base de données (ou carte de localisation) durant une phase d'apprentissage, en associant des signatures de signaux (par exemple RSSI) à des positions précises dans l'environnement. Lors de la phase d'exploitation, les mesures observées sont comparées à cette base pour estimer la position la plus probable. Cette méthode offre une bonne précision, notamment dans les environnements complexes, mais nécessite un effort initial de calibration.

- c. **Méthodes probabilistes** : Ces approches, comme les filtres de Kalman, les filtres particulaires, ou les modèles de Markov cachés, prennent en compte l'incertitude inhérente aux mesures et aux mouvements du dispositif. Elles permettent une localisation dynamique et robuste en intégrant des modèles statistiques sur les déplacements et les erreurs de capteurs. Elles sont particulièrement adaptées aux systèmes en temps réel [14].
- d. **Apprentissage automatique** : Les techniques d'apprentissage supervisé ou non supervisé, telles que les réseaux de neurones, les forêts d'arbres de décision ou les SVM (machines à vecteurs de support), permettent d'extraire des modèles complexes à partir de données de capteurs. Elles s'adaptent bien aux environnements variables, mais nécessitent des jeux de données d'entraînement riches et diversifiés. L'apprentissage profond, en particulier, ouvre la voie à des systèmes auto-adaptatifs pour la localisation indoor [15].

### 1.3 Systèmes de localisation avec LoRa

Plusieurs travaux en environnements (figure 1.4 [15]) exploitent les différences temporelles d'arrivée (TDoA), utilisant les horodatages des passerelles LoRa pour estimer la position d'un dispositif à partir des délais de réception entre différents récepteurs [16] [17] [18]. Cependant, ces méthodes présentent une précision limitée, principalement en raison de la faible résolution temporelle des horodatages ( $\approx 1 \mu s$ ). Étant donné la vitesse de propagation des ondes radio ( $\sim 300 \text{ m}/\mu s$ ), cette imprécision empêche une localisation fine [19]. Bien que des techniques de traitement du signal, comme les filtres de Kalman [20] [21], aient été proposées pour atténuer les effets du bruit, la résolution temporelle demeure une contrainte majeure.



**Figure 0.4** : Environnement intérieure (zone fermée).

D'autres travaux s'orientent vers l'exploitation du RSSI pour estimer les distances via des modèles de propagation, comme le proposent Lam et al [22], ou en sélectionnant dynamiquement les nœuds les moins bruités [23]. Récemment, des méthodes d'apprentissage automatique ont été intégrées à ces approches pour améliorer la robustesse. Aernouts et al [24]. ont utilisé l'algorithme kNN sur les mesures RSSI, tandis que Zhe et al [25]. ont modélisé ces mesures via un processus gaussien combiné à une estimation du maximum de vraisemblance.

D'autres études ont poursuivi cette direction [26] : applique divers algorithmes d'apprentissage automatique (régression par vecteurs de support, arbres de décision, modèles spline, etc.) à la localisation RSSI, bien que principalement pour des scénarios statiques [27] ; propose une méthode de fingerprinting combinée à une optimisation par essaim de particules et un raffinement bayésien, mais nécessitant un important prétraitement [28] ; intègre un modèle log-normal avec filtrage des valeurs aberrantes et prédiction par réseau de neurones à rétropropagation (BPNN), dépendant toutefois fortement de la qualité des données d'entraînement [29] ; développe un système de localisation intelligent et peu énergivore reposant sur des réseaux neuronaux aléatoires (RNN) et des données RSSI LoRa [30] ; introduit un système de fingerprinting combinant autoencodeur de débruitage et interpolations, avec des modèles profonds comme ANN, LSTM et CNN, montrant une précision supérieure aux approches classiques, mais demandant des mises à jour fréquentes des données interpolées [31] ; optimise le processus de collecte d'empreintes via une interpolation par forêt aléatoire, comparée à des algorithmes de correspondance par distance euclidienne minimale, avec de bons résultats mais une sensibilité accrue aux environnements dynamiques. Dans [15], un système de localisation haute précision est proposé en combinant RSSI et SNR, traités dans un système centralisé par réseaux neuronaux, bien que l'efficacité du SNR soit limitée dans les environnements à vue directe. Le modèle DeepFi-LoRaIn [32] intègre fingerprinting RSSI et apprentissage profond pour améliorer la localisation en environnements dynamiques, en tenant compte des interférences et variations, mais au prix de fréquents réentraînements. Enfin, l'étude [33] introduit la méthode ERSS, utilisant l'autocorrélation des frontières pour stabiliser la base de données d'empreintes et réduire la complexité de recherche, mais nécessitant un nombre important de points de référence, ce qui limite son évolutivité.

## **1.4 Conclusion**

Ce chapitre a présenté un panorama complet des systèmes de localisation en intérieur, en mettant en évidence la diversité des technologies disponibles, telles que le Wi-Fi, le BLE, le ZigBee, l'UWB ou encore LoRa. Ces systèmes s'appuient sur différentes méthodes d'estimation de position, reposant sur des grandeurs physiques comme le RSSI, l'AoA, le ToA ou le TDoA, et utilisent des algorithmes allant de la trilatération géométrique au fingerprinting, jusqu'aux approches avancées basées sur des modèles probabilistes ou d'apprentissage automatique.

L'analyse comparative des technologies et des méthodes a permis de souligner leurs avantages, limitations et domaines d'application respectifs, montrant que le choix d'une solution dépend fortement du contexte d'usage, des contraintes environnementales et des exigences de précision et d'autonomie.

Dans la continuité de cet état de l'art, ce travail se concentre sur l'exploitation du RSSI, en tant qu'indicateur facilement accessible, pour la localisation en intérieur. Afin de compenser les limitations inhérentes à cette métrique ; notamment sa sensibilité aux interférences et aux variations environnementales ; une approche basée sur la machine learning est envisagée. L'objectif est de tirer parti de la capacité des algorithmes d'apprentissage à modéliser les non-linéarités et à extraire des relations complexes entre le signal et la position, pour proposer une méthode de localisation plus précise, robuste et adaptable à divers environnements intérieurs.

## **CHAPITRE -2 :**

# **CADRE TECHNOLOGIQUE ET ALGORITHMIQUE POUR LA** **LOCALISATION EN INTERIEUR PAR RSSI ET LORA**

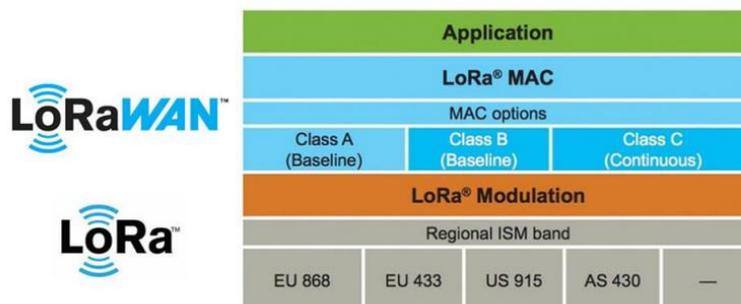
## 2.1 Introduction

A l'issue du premier chapitre, où nous avons présenté l'état de l'art des systèmes de localisation indoor, ainsi que les technologies et méthodes les plus couramment utilisées, nous avons pu constater que le choix d'une solution adaptée repose sur un compromis entre précision, portée, consommation d'énergie, coût et complexité d'implémentation.

Dans cette optique, ce chapitre vise à détailler les fondements technologiques et méthodologiques de la solution retenue dans ce mémoire, qui s'appuie sur la technologie LoRa/LoRaWAN pour la transmission des données et sur le RSSI (Received Signal Strength Indicator) comme paramètre principal pour l'estimation de la position. En raison des limites inhérentes à la précision du RSSI dans des environnements complexes, l'intégration de techniques d'apprentissage automatique est explorée afin d'améliorer la robustesse et la fiabilité du système de localisation.

## 2.2 Technologie Lora/LoRaWAN

Initialement développée par des ingénieurs de la start-up française Cycleo, fondée à Grenoble en 2009, avant d'être acquise par l'entreprise Semtech en 2012, LoRa, *Long Range*, est une technologie de communication sans fil appartenant à la famille des réseaux LPWAN (*Low Power Wide Area Network*), conçue pour assurer une transmission de données à longue portée avec une faible consommation d'énergie [34].



**Figure 0.1** : LoRa et LoRaWAN.

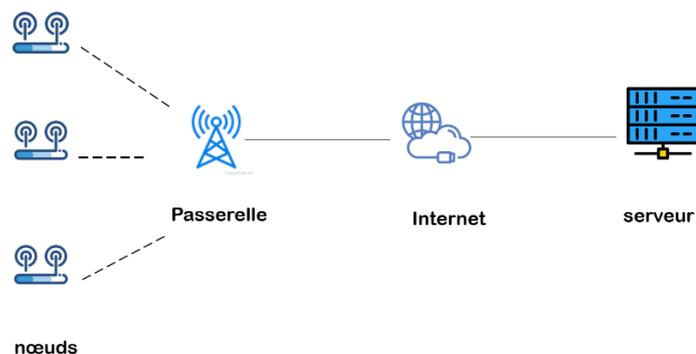
Elle repose sur une modulation d'ondes radio spécifique et constitue la base physique des réseaux LoRaWAN (Long Range Wide Area Network) qui se distingue par ses faibles débits de données, atteignant un maximum de 27 kb/s, ce qui est compensé par une portée

importante et une consommation énergétique très faible. Elle supporte des largeurs de bande de 125, 250 ou 500 kHz, choisies selon le compromis souhaité entre portée, robustesse et débit. LoRa fonctionne dans les bandes de fréquence ISM (libres de licence), avec des plages régionales spécifiques : 433 MHz en Asie, 868 MHz en Europe, et 915 MHz en Amérique [35]. La figure 2.1[36] illustre la différence entre LoRa et LoRaWAN.

## 1. Réseau LoRaWAN

LoRaWAN constitue la couche de contrôle d'accès au support (MAC) du protocole LoRa, définissant les règles de communication, de gestion des données et de sécurité au sein du réseau. Son architecture repose sur une topologie en étoile, comme illustré à la figure 2.2, où une ou plusieurs passerelles servent de relais entre les dispositifs terminaux (capteurs ou actionneurs) et le serveur réseau.

Dans ce modèle, les terminaux communiquent uniquement avec les passerelles, sans connexion directe entre eux. Les passerelles, quant à elles, transmettent les messages reçus vers le serveur réseau via Internet, souvent en utilisant des protocoles IP standards. Le serveur réseau filtre, déduplique et gère les messages, avant de les transmettre, le cas échéant, au serveur d'application chargé d'exploiter les données dans un contexte spécifique (ex. localisation, suivi de santé, domotique, etc.) [37].



**Figure 0.2** : Architecture de LoRaWan.

Cette organisation (figure 2.2) favorise la scalabilité, la faible consommation énergétique, ainsi que la portée étendue, faisant de LoRaWAN une infrastructure bien adaptée aux réseaux IoT massifs, notamment pour des applications de localisation indoor et outdoor [38].

LoRaWAN définit également trois classes de dispositifs (A, B et C), permettant d'adapter le comportement de communication à la nature de l'application, en particulier en termes de latence et de consommation énergétique :

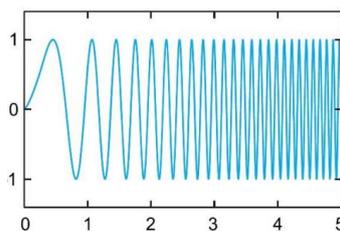
Classe A (par défaut) : tous les dispositifs LoRaWAN doivent implémenter cette classe. Elle offre le meilleur compromis énergétique, car chaque terminal ouvre une courte fenêtre de réception uniquement après une émission, ce qui réduit considérablement la consommation. C'est la classe la plus adaptée aux objets autonomes alimentés sur batterie.

Classe B : en plus du comportement de la classe A, les dispositifs de classe B reçoivent des signaux de synchronisation (beacons) envoyés par la passerelle à intervalles réguliers. Cela permet au serveur réseau de planifier les messages descendants à des moments précis, réduisant ainsi la latence tout en conservant une consommation modérée.

Classe C : les dispositifs de cette classe maintiennent leur récepteur actif en permanence, sauf lors d'une émission. Cela permet de recevoir des données à tout moment, avec une latence minimale, mais au prix d'une consommation énergétique élevée. Elle convient principalement aux équipements alimentés en continu (par exemple sur secteur) [39].

## 2. Modulation

La modulation LoRa repose sur une technique d'étalement du spectre par Chirp (*Chirp Spread Spectrum – CSS*), dans laquelle chaque symbole est représenté par un Chirp à fréquence linéairement variable. Comme cela est visualisé dans la figure 2.3 [40] . Cette méthode assure une bonne robustesse face aux interférences et au bruit. Chaque symbole encode un nombre de bits déterminé par le facteur d'étalement (*Spreading Factor – SF*), dont les valeurs varient entre 7 et 12, avec  $2^{SF}$  chips par symbole. Une valeur élevée de SF améliore la portée et la résistance au bruit, mais réduit le débit de données. Les SF étant orthogonaux, plusieurs nœuds peuvent émettre simultanément sur une même fréquence sans causer d'interférences mutuelles, ce qui optimise l'utilisation du canal radio [38].



**Figure 0.3** : Visualisation des chirps dans la modulation LoRa.

LoRa adapte dynamiquement le SF en fonction du rapport signal/bruit (SNR) et de la largeur de bande, afin de maintenir une communication fiable. En complément, un schéma d'étalement par saut de fréquence (*FHSS*) est utilisé, modifiant aléatoirement la fréquence de transmission pour limiter les interférences. De plus, la modulation CSS utilisée génère un pic d'autocorrélation net, utile pour identifier précisément le moment de réception d'un signal.

Cette propriété est précieuse dans les applications de localisation, puisqu'elle permet d'estimer avec précision le temps de propagation entre deux nœuds, principe également exploité dans certains systèmes radar [38].

### **2.3 L'indicateur De Puissance Du Signal Reçu (RSSI)**

L'indicateur de puissance du signal reçu, ou RSSI, est une mesure de la puissance d'un signal radio, lorsqu'il est capté par un récepteur, tel qu'un module LoRa ou une passerelle. Dans les systèmes de communication sans fil, le RSSI constitue une information essentielle, car il reflète l'atténuation subie par le signal lors de sa transmission dans l'air.

Lorsque les signaux radio se propagent, ils sont soumis à divers phénomènes physiques (réflexion, diffraction, absorption, interférences), ce qui entraîne une perte de puissance, appelée atténuation. Cette perte est décrite par des modèles mathématiques appelés modèles de propagation [41].

#### **a. Modèle de perte en espace libre (Free Space Path Loss - FSPL)**

Dans un environnement idéal avec une ligne de vue directe (LOS),[42] la puissance reçue  $P_r$  est liée à la puissance transmise  $P_t$  selon le modèle FSPL :

$$P_r [dBm] = P_t [dBm] - 20 \log_{10} (d) - 20 \log_{10} (4\pi / \lambda) \quad \dots (2.1)$$

Où :

- $d$  est la distance entre l'émetteur et le récepteur,
- $\lambda$  est la longueur d'onde du signal.

Ce modèle montre que le RSSI diminue logarithmiquement avec la distance : plus le récepteur est éloigné, plus le signal est faible.

### **b. Modèle log-distance**

Dans des conditions réelles (intérieur, ville, obstacles), le modèle FSPL est étendu pour intégrer l'environnement grâce à un exposant d'atténuation  $\gamma$ . Le RSSI est alors exprimé par :

$$RSSI = Pr(d) = P_t - PL_0 - 10\gamma \log_{10}(d_0/d) \quad \dots (2.2)$$

Où :

- $PL_0$  est la perte mesurée à une distance de référence  $d_0$ ,
- $\gamma$  est un paramètre variant selon l'environnement (typiquement entre 2 et 4).

### **c. Effet d'ombrage (Shadowing)**

Les phénomènes imprévisibles comme les interférences, les réflexions ou les conditions météo créent des fluctuations autour de la valeur théorique du RSSI. On ajoute donc un terme aléatoire  $X_g$ , souvent modélisé par une variable gaussienne à moyenne nulle :

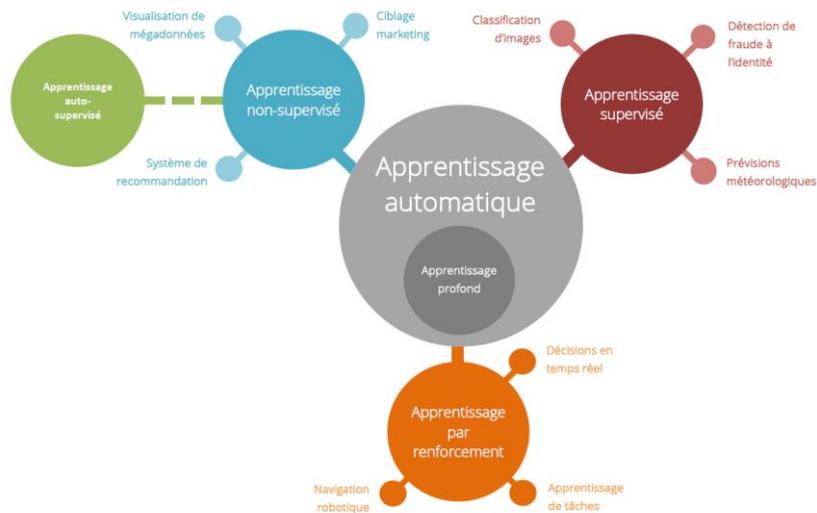
$$RSSI = P_t - PL_0 - 10\gamma \log_{10}(d_0/d) + X_g \quad \dots (2.3)$$

Ce modèle montre que la relation entre RSSI et distance est bruitée et non parfaitement déterministe.

## **2.4 L'apprentissage Automatique (Machine Learning)**

L'apprentissage automatique est un pilier fondamental de nombreuses applications dans l'Internet des objets (IoT), notamment pour la prise de décision intelligente à partir de données collectées. Il vise à doter les systèmes informatiques de la capacité à apprendre de leurs expériences et à améliorer leurs performances sans programmation explicite.

Le processus d'apprentissage repose généralement sur trois étapes clés : la collecte et le traitement de données, l'entraînement d'un modèle sur ces données, puis l'évaluation de ses performances sur un jeu de test. Ce modèle peut ensuite être utilisé pour effectuer des prédictions ou classer de nouvelles données [15].



**Figure 0.4** : Types d'apprentissage automatique.

On distingue principalement quatre types d'apprentissage automatique. Comme le montre la figure 2.4 [43]:

- **Apprentissage supervisé** : L'algorithme apprend à partir d'un ensemble de données étiquetées, où chaque entrée est associée à une sortie attendue.
- **Apprentissage non supervisé** : Il s'agit d'identifier des structures ou des regroupements dans des données non étiquetées (clustering).
- **Apprentissage semi-supervisé** : Combine les approches précédentes en exploitant à la fois des données étiquetées et non étiquetées.
- **Apprentissage par renforcement** : Le système apprend par essais-erreurs, en recevant une récompense ou une pénalité selon les actions entreprises.[15]

Dans ce travail, nous nous concentrons sur l'apprentissage supervisé, particulièrement adapté à la prédiction de localisation en intérieur à partir de données RSSI.

### 2.4.1 Algorithmes sélectionnés

Pour estimer la position à partir des mesures RSSI, nous utilisons trois algorithmes largement reconnus pour leur efficacité.

#### 1. K-plus proche voisin

L'algorithme des K-plus proches voisins (K-NN) est l'une des méthodes de classification les plus simples et les plus couramment utilisées en apprentissage automatique.

supervisé. Son principal avantage réside dans son approche non paramétrique, ce qui signifie qu'il ne repose sur aucune hypothèse a priori concernant la distribution des données. Cette flexibilité en fait un outil particulièrement adapté à des applications variées, y compris dans le domaine de la localisation indoor, où les données peuvent présenter des structures complexes.

Contrairement à de nombreux autres algorithmes, le K-NN ne nécessite pas de phase d'entraînement explicite : il est qualifié d'"apprentissage paresseux" (lazy learning) car il diffère tout traitement jusqu'à la phase de prédiction. Au lieu de construire un modèle généralisé, il stocke l'ensemble des données d'apprentissage et effectue les classifications en temps réel en comparant chaque nouvelle observation aux échantillons existants [44].

Le K-NN repose sur deux éléments principaux :

- Une mesure de similarité (distance) pour évaluer la proximité entre les points de données.
- Le paramètre  $k$ , qui détermine le nombre de voisins pris en compte pour la décision de classification.

Les données doivent être représentées dans un espace métrique, où les notions de distance sont bien définies. Elles peuvent être de nature scalaire ou vectorielle (multidimensionnelle).

Plusieurs métriques peuvent être employées pour calculer les distances entre points :

- **Distance euclidienne ( $L_2$ ) :**

$$D_e(x, y) = \sqrt{\sum_{j=1}^n (x_j - y_j)^2} \quad \dots (2.4)$$

La distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points. Où :

- $D_e(x, y)$  : distance euclidienne entre les vecteurs  $x$  et  $y$
- $n$  : nombre total de composantes (ou dimensions) du vecteur
- $x_j, y_j$  : composante n°  $j$  des vecteurs  $x$  et  $y$

- **Distance de Manhattan ( $L_1$ ) :**

$$D_m(x, y) = \sqrt{\sum_{i=1}^k |x_i - y_i|} \quad \dots (2.5)$$

La distance qui Calcule la somme des valeurs absolues des différences entre les coordonnées de deux points. Où :

- $D_m(x, y)$  : distance de Manhattan entre  $x$  et  $y$
- $k$  : nombre total de dimensions (pareil que  $n$ )
- $x_i, y_i$  : composante n°  $i$  des vecteurs

- **Distance de Tchebychev ( $L_\infty$ ) :**

$$D_h(x, y) = \max_i |x_i - y_i| \quad \dots (2.6)$$

Elle mesure la différence maximale entre les coordonnées, utile dans des espaces multidimensionnels où certaines dimensions dominent. Où :

- $D_h(x, y)$  :
- $x_i, y_i$  : composantes des vecteurs

En gros, l'algorithme KNN mesure les distances entre l'instance à tester et tous les points de données, choisit les  $k$  voisins les plus proches, puis attribue l'étiquette qui est la plus fréquente parmi ces voisins comme étiquette prédite pour l'instance à tester [44].

## 2. Machine a vecteur de support

La machine à vecteurs de support (SVM) est un algorithme d'apprentissage supervisé largement utilisé pour les tâches de classification binaire. Son objectif principal est de déterminer une frontière de décision optimale (hyperplan) qui sépare les données en deux classes tout en maximisant la marge entre elles. Cette approche permet d'améliorer la généralisation du modèle et de réduire les risques de surapprentissage.

Le SVM cherche à trouver l'hyperplan qui maximise la distance entre les points les plus proches de chaque classe, appelés vecteurs de support. Ces points jouent un rôle crucial dans la définition de la frontière de décision, car ils déterminent la position et l'orientation de l'hyperplan [45].

- **Paramètres clés des SVM**

Les performances d'un modèle SVM dépendent de plusieurs paramètres essentiels :

- Marge maximale La marge correspond à la distance entre l'hyperplan et les vecteurs de support les plus proches.
- Une marge plus grande améliore la capacité de généralisation du modèle en réduisant le surapprentissage.

- **Fonctions de noyau (Kernel)**

Les SVM peuvent utiliser différentes fonctions de noyau pour transformer les données dans un espace de dimension supérieure, permettant ainsi une séparation linéaire même lorsque les données ne le sont pas dans l'espace d'origine. Les noyaux couramment utilisés sont :

- Linéaire : Pour des données linéairement séparables.
- Polynomial : Pour des relations non linéaires.
- Radial Basis Fonction (RBF) : Adapté aux problèmes complexes avec des frontières non linéaires.
- Sigmoidale : Principalement utilisé dans des cas spécifiques comme les réseaux de neurones.

- **Paramètre de régularisation (C)**

C contrôle le compromis entre la maximisation de la marge et la minimisation des erreurs de classification.

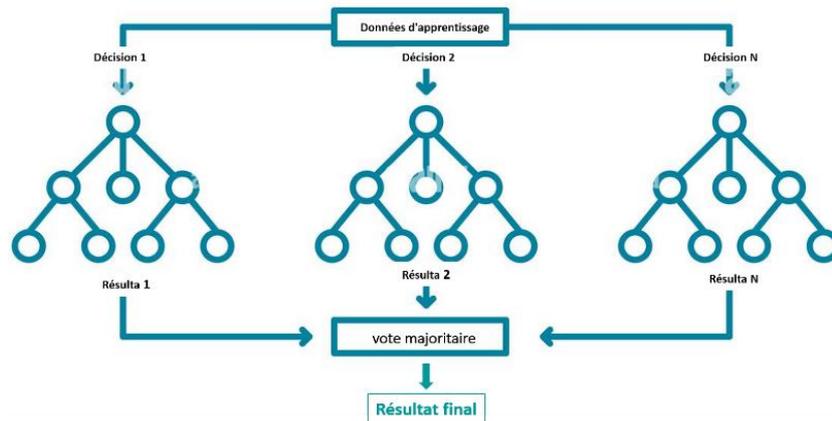
- Une valeur faible de C tolère davantage d'erreurs mais augmente la marge (moins de surapprentissage)
- Une valeur élevée de C réduit la marge pour minimiser les erreurs d'entraînement (risque de surapprentissage).

- **Paramètre Gamma ( $\gamma$ )**

Gamma influence l'étendue de l'impact d'un seul point de données sur la frontière de décision.

- Une valeur faible de  $\gamma$  donne une frontière plus lisse, car elle considère des points éloignés.
- Une valeur élevée de  $\gamma$  rend la frontière plus complexe, car elle se concentre sur les points proches [45].

### 3. Forêts aléatoires



**Figure 0.5 :** Processus de Random forest.

Un arbre de décision est un modèle d'apprentissage automatique non paramétrique couramment utilisé pour des tâches de classification et de régression. Il repose sur une structure arborescente. Comme le montre la figure 2.5 [46], où chaque nœud représente une règle de décision fondée sur les caractéristiques des données en entrée.

Cette structure hiérarchique se compose de plusieurs éléments clés :

- Le nœud racine constitue le point de départ de l'arbre. Il contient l'ensemble des données d'apprentissage et initie le processus de partitionnement.
- Les nœuds de décision représentent les points où les données sont divisées selon des conditions logiques basées sur les attributs.
- Les branches correspondent aux différentes issues possibles des tests réalisés dans les nœuds de décision, conduisant vers d'autres nœuds.
- Les nœuds feuilles représentent les résultats finaux. En classification, ils indiquent une étiquette de classe, tandis qu'en régression, ils donnent une valeur numérique prédite.

- Le fractionnement désigne le processus de division d'un nœud parent en plusieurs nœuds enfants.
- L'élagage consiste à supprimer certaines branches de l'arbre afin de limiter sa complexité et d'éviter le surapprentissage.
- Les relations parent/enfant désignent le lien hiérarchique entre un nœud supérieur (parent) et les nœuds issus de sa division (enfants).[45]

L'algorithme commence par évaluer la capacité de chaque caractéristique à séparer efficacement les données, en utilisant des critères tels que l'indice de Gini ou le gain d'information. L'indice de Gini est défini comme suit :

$$G(t) = 1 - \sum_{i=1}^k p_i^2 \quad \dots (2.7)$$

- $G(t)$ : indice de Gini pour le nœud t
- $k$ : nombre total de classes
- $p_i$  : Proportion d'exemples de la classe  $i$  dans le nœud t

La caractéristique offrant la meilleure séparation est choisie comme racine, et le jeu de données est divisé en sous-ensembles. Ce processus récursif se poursuit à chaque nœud, générant progressivement la structure de l'arbre, jusqu'à ce qu'un critère d'arrêt soit atteint — par exemple, une profondeur maximale prédéfinie ou un nombre minimal d'observations dans un nœud.

Dans le cadre de la classification, l'arbre est parcouru selon les valeurs des caractéristiques d'une nouvelle instance, jusqu'à ce qu'un nœud feuille soit atteint, fournissant ainsi l'étiquette prédite. En régression, la feuille renvoie une estimation numérique calculée à partir des valeurs cibles présentes dans le sous-ensemble correspondant.

Les arbres de décision se distinguent par leur interprétabilité : les règles de décision peuvent être facilement visualisées, rendant le modèle compréhensible même pour des non-spécialistes. Ils sont également capables de gérer des données hétérogènes (catégorielles et numériques) et sont relativement robustes aux valeurs manquantes ou aberrantes. Toutefois,

s'ils ne sont pas régularisés, ils peuvent devenir trop complexes et suradapter les données d'entraînement.

Pour limiter ce phénomène de surapprentissage, plusieurs techniques peuvent être employées, telles que l'élagage, la limitation de la profondeur de l'arbre, ou encore le recours à des approches d'ensemble comme les forêts aléatoires (Random Forest) ou le boosting, qui combinent plusieurs arbres pour renforcer la performance et la généralisation [47].

Une fois tous les arbres construits, chaque arbre effectue une prédiction  $H_t(x)$ , et la classe  $y$  est déterminée par un vote majoritaire :

$$y = \text{mode}\{H_1(x), H_1(x), \dots, H_T(x)\} \quad \dots (2.8)$$

- $y$  : prédiction finale du modèle Random Forest
- $H_t(x)$  : prédiction du classifieur (arbre) numéro  $t$  pour l'entrée  $x$
- $T$  : nombre total d'arbres dans la forêt
- *mode* : opérateur qui retourne la classe la plus fréquente parmi les prédictions des  $T$  arbres (vote majoritaire)

De plus, Random Forest fournit des mesures d'importance des caractéristiques, permettant d'identifier les variables les plus influentes dans la classification. Belgiu et Dragut (2016) ont démontré que Random Forest est particulièrement performant dans la classification des données spatiales.

## 2.5 Conclusion

Dans ce chapitre, nous avons établi les bases techniques nécessaires à la compréhension de la localisation intérieure par LoRaWAN. L'étude du RSSI et de son modèle d'atténuation a permis de souligner les défis liés à la précision des mesures en milieu intérieur. Par ailleurs, l'intégration d'algorithmes d'apprentissage automatique tels que KNN, Random Forest et SVM apparaît comme une solution pertinente pour optimiser la localisation. Les concepts abordés ici serviront de fondement à la méthodologie de mise en œuvre et d'évaluation du système, qui sera présentée dans les chapitre suivants.

**CHAPITRE -3 :**

**IMPLEMENTATION MATERIELLE ET LOGICIEL**

### 3.1 Introduction

Dans ce chapitre, nous examinons la conception et l'implémentation d'un système de localisation intérieure reposant sur la technologie LoRa. Nous précisons les choix technologiques réalisés, en mettant en avant les modules matériels employés pour la collecte et la transmission des données. Par ailleurs, nous clarifions le fonctionnement du système en décrivant les interactions entre l'émetteur et le récepteur, ainsi que les facteurs qui impactent la performance du réseau LoRa.

### 3.2 Schéma Synoptique Du Système :

La figure 3.1 illustre le schéma synoptique de notre solution. L'architecture matérielle de notre système de localisation est composée d'un nœud mobile et d'une passerelle communiquant via le protocole LoRa, tandis que le traitement des données est effectué hors ligne.

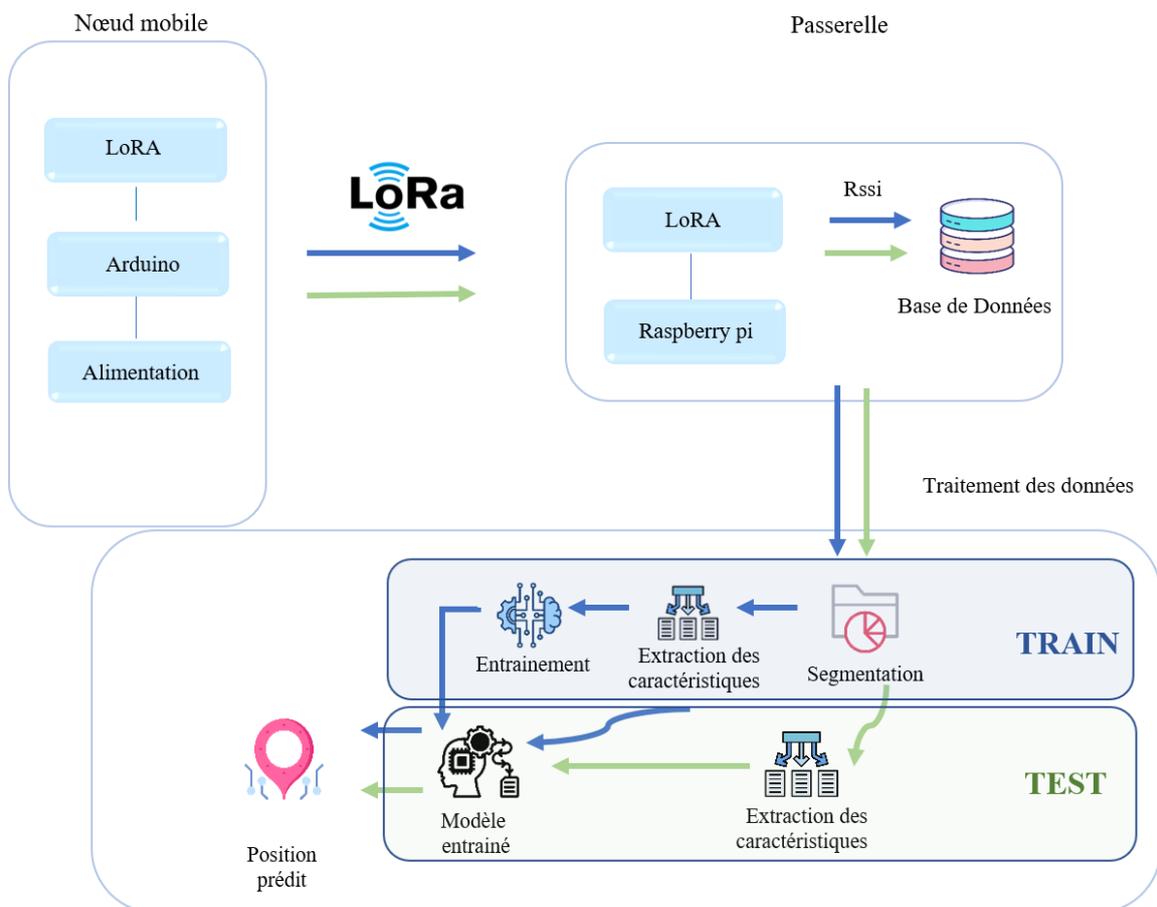
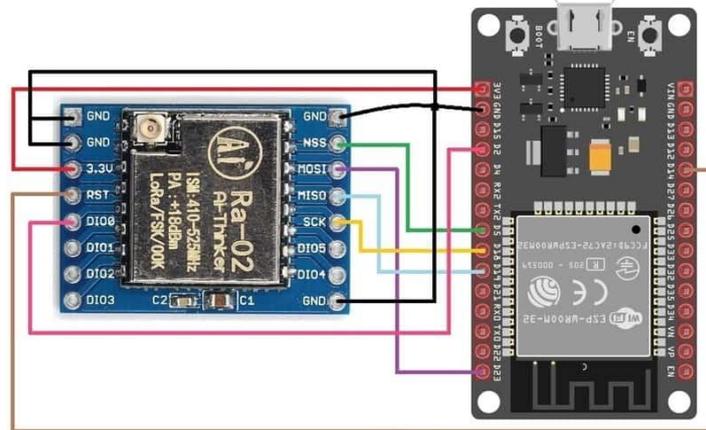


Figure 0.1 : Schéma synoptique du système.

### 3.2.1 Architecture Matériel de la solution

#### 3.2.1.1 Nœud Mobile

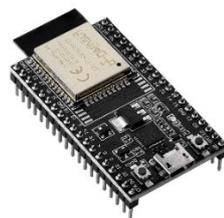
Notre nœud mobile est construit autour d'un microcontrôleur ESP32 et d'un module LoRa, dont le schéma électrique est illustré à la figure 3.2 [48].



**Figure 0.2** : Schéma électrique du Nœud mobile.

#### a. ESP32

L'ESP32 est un microcontrôleur conçu par Expressif Systems (figure 3.3) [49], qui intègre un processeur dual-core Tensilica Xtensa LX6 fonctionnant à des fréquences allant jusqu'à 240 MHz.



**Figure 0.3** : Module ESP32.

Il offre des fonctionnalités avancées telles que le Wi-Fi, le Bluetooth Low Energy (BLE) et plusieurs interfaces de communication (SPI, I2C, UART, ADC, DAC, PWM).

Conçu pour les applications IoT (Internet des Objets), il se distingue par sa faible consommation d'énergie grâce à ses modes d'économie d'énergie, ce qui en fait un choix idéal pour les systèmes embarqués, la domotique et les solutions de communication sans fil à longue portée, comme LoRa, lorsqu'il est associé à un module radio externe [50].

## b. Module LoRa

Les modules SX1276/77/78/79 (figure 3.4) [51] intègrent un modem LoRa conçu pour les communications longue portée, basé sur une technique de spectre étalé.



**Figure 0.4** : Module LoRa.

Cette technologie permet des transmissions sur de grandes distances tout en offrant une excellente résistance aux interférences et une consommation énergétique réduite.

Grâce à la modulation LoRa brevetée par Semtech, ces modules peuvent atteindre une sensibilité remarquable, supérieure à -148 dBm, tout en utilisant des composants simples tels qu'un cristal et des matériaux économiques.

Ils intègrent également un amplificateur de puissance capable de délivrer jusqu'à +20 dBm, ce qui leur confère l'un des meilleurs budgets de liaison du marché. Cette performance en fait une solution idéale pour les applications exigeant à la fois grande portée et robustesse de communication [52].

## c. Logiciels et bibliothèques utilisés

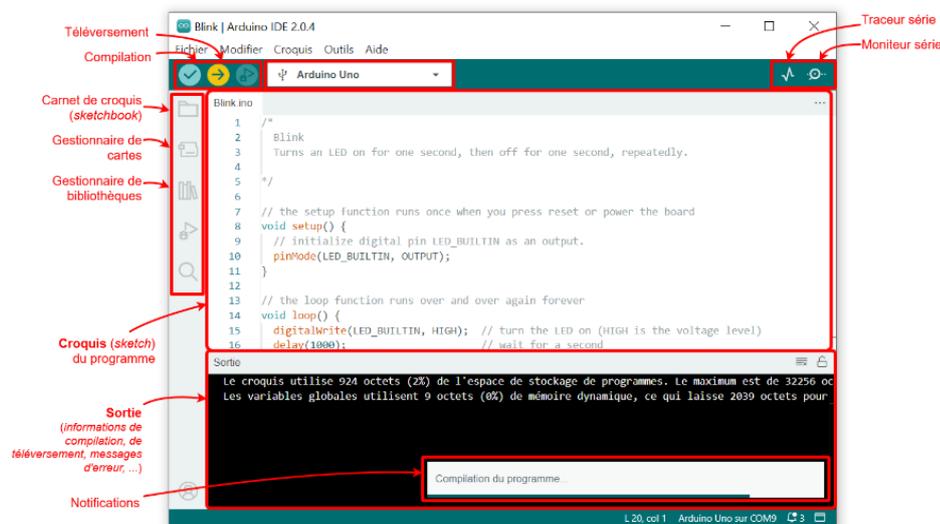
Afin de programmer l'ESP32 et d'assurer le bon fonctionnement du nœud mobile, l'environnement de développement Arduino IDE ainsi que des bibliothèques spécifiques ont été utilisés.

### ➤ Arduino IDE

L'Arduino IDE (figure 3.5) est un environnement de développement open-source et gratuit [53], largement utilisé dans le domaine de l'embarqué. Il offre plusieurs fonctionnalités essentielles :

- **Édition du code** : L'IDE permet d'écrire des programmes (appelés *sketches*) en langage C/C++.

- **Compilation** : Le code est compilé en langage compréhensible par le microcontrôleur. La zone de sortie affiche les messages de progression, les erreurs éventuelles et les notifications utiles à la correction du code.
- **Téléversement** : Une fois compilé, le programme est transféré vers l'ESP32 via une connexion USB. Une fois en mémoire, le programme devient un microgiciel (*firmware*).
- **Communication série** : L'IDE propose également un moniteur série permettant d'échanger des données entre la carte et l'ordinateur, utile pour le débogage ou l'affichage des mesures en temps réel tant que la connexion physique est maintenue.



**Figure 0.5** : Arduino IDE.

### ➤ Bibliothèque SPI

La bibliothèque SPI est utilisée pour établir une communication entre le microcontrôleur (comme l'ESP32) et des périphériques externes via le protocole SPI (Serial Peripheral Interface). Ce protocole de communication série permet un échange rapide de données avec des composants tels que des capteurs, des écrans ou des modules de stockage [54]. Dans notre cas, cette bibliothèque a été employée pour assurer la liaison entre le microcontrôleur et le module LoRa.

### ➤ La bibliothèque LoRa

La bibliothèque LoRa met à disposition un ensemble de fonctions et de méthodes prédéfinies facilitant l'établissement de la communication entre le microcontrôleur et le module LoRa. Elle simplifie l'envoi et la réception de données en masquant la complexité sous-jacente du protocole [55].

### 3.2.1.2 Passerelle

Le schéma électrique de la passerelle, construite autour d'un Raspberry Pi 4 associé à un module LoRa, est présenté à la figure 3.6.

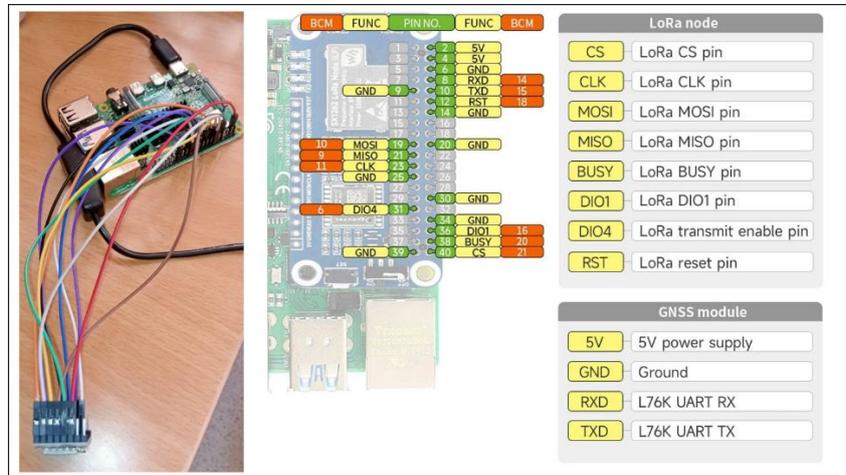


Figure 0.6 : Schéma électrique de la passerelle.

#### a. Raspberry Pi

Le Raspberry Pi (figure 3.7) [56] est un micro-ordinateur compact aux performances remarquables, développé par la Fondation Raspberry Pi, une organisation caritative basée à Cambridge (Royaume-Uni). Conçu en 2009 par Eben Upton et son équipe, son objectif initial était de démocratiser l'accès à l'informatique, notamment auprès des jeunes, afin de les initier à la programmation et aux sciences informatiques [57].

Fonctionnant comme un ordinateur classique, le Raspberry Pi utilise généralement un système d'exploitation basé sur Linux, installé sur une carte SD. Il peut être connecté à un écran, un clavier et une souris, et exécuter des tâches comparables à celles d'un PC ou d'un petit serveur [57].

Le Raspberry Pi 4, dernière génération à ce jour, présente des spécifications proches de celles d'un ordinateur d'entrée de gamme. Équipé de jusqu'à 8 Go de RAM et d'un processeur performant, il constitue une alternative sérieuse pour les usages courants. Sa richesse en interfaces (ports GPIO, HDMI, USB, caméra, etc.) lui permet également de répondre à un large éventail d'applications, notamment dans les domaines de l'IoT, de la domotique ou de la robotique [58].

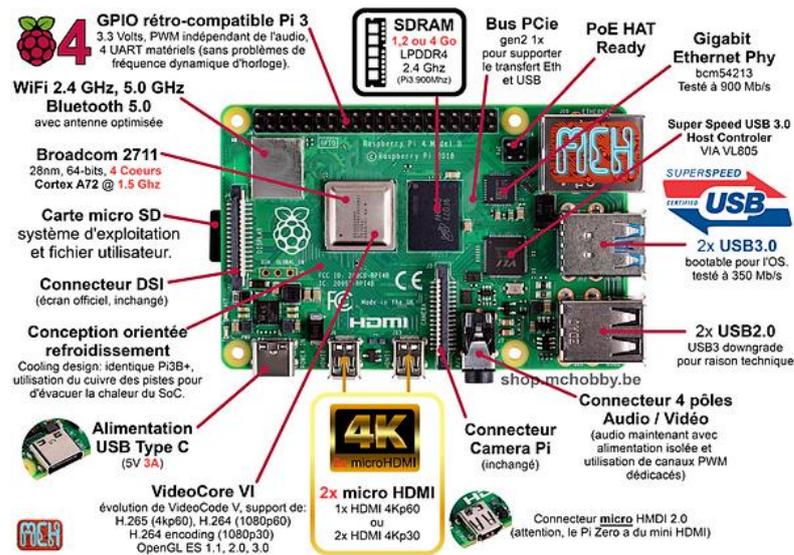


Figure 0.7 : Raspberry Pi4.

Ses caractéristiques techniques principales sont les suivantes :

- **Microprocesseur** : Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit à 1,5 GHz
- **Mémoire vive** : LPDDR4 de 1 Go, 2 Go, 4 Go ou 8 Go selon le modèle
- **Connectivité** :
  - Wi-Fi double bande (2,4 GHz et 5,0 GHz) – IEEE 802.11 b/g/n/ac
  - Bluetooth 5.0, BLE
- **Interfaces** :
  - Connecteur GPIO 40 broches (compatible avec les générations précédentes)
  - 2 ports micro HDMI (jusqu'à 4Kp60)
  - Port d'affichage MIPI DSI pour écran LCD
  - Port caméra MIPI CSI
  - Port Gigabit Ethernet
  - 2 ports USB 3.0 et 2 ports USB 2.0
  - Jack 4 pôles : sortie vidéo composite + audio stéréo
- **Multimédia** :
  - Décodage H.265 (4Kp60)
  - Décodage H.264 (1080p60) / Encodage H.264 (1080p30)
  - Support OpenGL ES 3.0
- **Stockage** : Connecteur pour carte microSD (utilisé pour l'OS et les données)
- **Alimentation** : 5V DC via connecteur USB-C (min. 3A)

## **b. Module LoRa**

Le module LoRa utilisé avec le Raspberry Pi est identique à celui intégré dans le nœud mobile, à savoir le module SX1276. Ce choix garantit une compatibilité parfaite et une cohérence dans les performances de communication entre les deux entités du système.

## **c. Logiciels et bibliothèques utilisés**

Afin d'assurer le bon fonctionnement de la passerelle, plusieurs logiciels et bibliothèques ont été utilisés, parmi lesquels :

### ➤ Thonny

Thonny est un IDE pour le langage Python, particulièrement adapté aux débutants. Il est préinstallé sur le système d'exploitation du Raspberry Pi, ce qui permet de démarrer rapidement le développement sans configuration préalable.

Son interface épurée et intuitive facilite la prise en main. Thonny intègre un interpréteur Python permettant d'exécuter directement les scripts, et propose de nombreuses fonctionnalités utiles pour l'apprentissage et le débogage. L'une de ses forces réside dans son mode de débogage pas à pas, qui permet d'analyser le programme ligne par ligne. Cela offre une visualisation claire de la création des variables, des objets, et du passage des valeurs entre les fonctions ou comparateurs [59].

Thonny est accessible directement depuis le bureau de l'environnement graphique du Raspberry Pi.

### ➤ Python

Python est un langage de programmation de haut niveau interprété pour la programmation à usage général. Créé par Guido van Rossum et publié pour la première fois en 1991, Python repose sur une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces significatifs. Il fournit des constructions permettant une programmation claire à petite et grande échelle. Python propose un système de typage dynamique et une gestion automatique de la mémoire. Il prend en charge plusieurs paradigmes de programmation, notamment orienté objet, impératif, fonctionnel et procédural, et dispose d'une bibliothèque standard étendue et complète [60].

### ➤ Bibliothèque RPI.GPIO

La bibliothèque RPi.GPIO Python permet de configurer facilement et de lire/écrire les broches d'entrée / sortie sur l'en-tête GPIO dans un script Python [59]. Pour installer la bibliothèque RPi.GPIO sur un Raspberry Pi, nous utilisons la commande suivante :

- Pip Install RPi.GPIO
  - Bibliothèque Pylora

Pylora est un module Python qui permet la transmission et la réception des données à l'aide d'un émetteur-récepteur LoRa, en particulier le modèle SX127\_ ICS, développé par Semtech [61]. Pour installer la bibliothèque, nous utilisons la commande suivante :

- Pip Install Pylora
  - Bibliothèque Sys

La bibliothèque Sys de Python fournit l'accès à certaines variables utilisées par l'interpréteur pour manipuler l'environnement d'exécution Python [62].

- Bibliothèque struct

Cette bibliothèque permet de manipuler et d'interpréter des données binaires structurées, notamment pour l'encodage et le décodage de valeurs de différents types de données [63].

- Bibliothèque Spidev

Spidev est une liaison Python pour Linux qui peut être utilisée pour effectuer des communications SPI sur Raspberry [64].

- Pip Install Spidev
  - Bibliothèque MariaDB Connector

Pour enregistrer les informations récupérées, nous utilisons une base de données MariaDB. Pour cela, Afin d'établir une connexion avec la base de données. Nous devons installer la bibliothèque du connecteur MariaDB. Le processus d'installation reste inchangé.

Le module MariaDB Connector fournit La méthode "Connect" qui utilisée pour établir une connexion entre deux objets, résultant en la création d'un objet qui représente la connexion, Nous devons spécifier l'hôte et le nom du domaine que nous voulons enregistrer, Utilisateur et mot de passe pour établir la connexion. De plus, nous pouvons également spécifier le paramètre "database" pour indiquer la base de données souhaitée [65].

#### **d. Base de données**

Pour assurer la conservation des données, une base de données locale a été déployée sur le Raspberry Pi. Elle permet d'enregistrer et de consulter les valeurs estimées de RSSI et de SNR.

Le choix s'est porté sur MariaDB, un système de gestion de base de données relationnelle open source, en raison de sa légèreté, de sa compatibilité avec MySQL, et de ses performances bien adaptées aux environnements embarqués.

Une base de données dédiée a été créée, incluant une table spécifique aux mesures. Cette table comprend les champs suivants :

- **Id** : identifiant unique
- **Timestamp** : date et heure de réception
- **Rssi** : valeur du signal reçu
- **Snr** : rapport signal/bruit

#### **3.2.1.3 Principe de fonctionnement de la solution**

Dans les deux programmes exécutés respectivement sur l'ESP32 et le Raspberry Pi, une phase d'initialisation est nécessaire. Celle-ci comprend la configuration des entrées/sorties, des bus de communication, ainsi que du module LoRa. Cette étape est indispensable avant tout échange de données. Le module LoRa a été configuré avec les paramètres suivants :

- **Bande passante (BW)** : 125 kHz
- **Fréquence** : 443 MHz, définissant la bande de spectre utilisée pour les communications
- **Facteur d'étalement (SF)** : fixé à 7, ce qui correspond à un étalement de 128 chips par symbole
- **Puissance d'émission (TX)** : 13 dBm, un compromis entre portée et consommation d'énergie
- **Taux de codage (CR)** : 4/5, permettant d'améliorer la robustesse face aux interférences

```

7 BOARD_setup()
8
9 class LoRaRecvCont(LoRa):
10 def __init__(self, verbose=False):
11     super(LoRaRecvCont, self).__init__(verbose)
12     self.set_mode(MODE_SLEEP)
13     self.set_dio_mapping(DIO1 = 0)
14     self.set_sync_word(0xF3) # Ensure the sync word matches the transmitter
15     self.set_freq(433) # Set frequency to 433 MHz
16     print("LoRa initialized")
17
18 def start(self):
19     self.reset_dio_rx()
20     self.set_mode(MODE_RXCONT)
21     print("Starting RX continuous mode")
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

Serial Monitor x Output
Message (Enter to send message to 'Arduino Uno' on 'COM3')
Sending packet: 5
Sending packet: 6
Sending packet: 7
Sending packet: 8
Sending packet: 9
Sending packet: 10
Sending packet: 11
Sending packet: 12
Sending packet: 13
Sending packet: 14

```

**Figure 0.8** : Message d'initialisation.

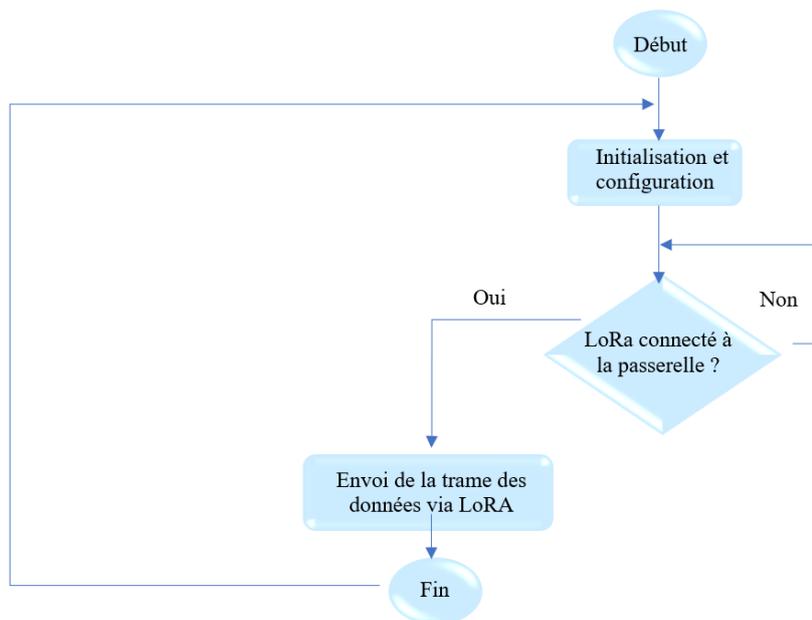
Une fois toutes les initialisations et configurations terminées, l'ESP32 envoie une trame intitulée « Hello » vers la passerelle Raspberry Pi (figure 3.8). Cette trame est structurée de la manière suivante :

Préambule	En-tête	Id	Timestamp	RSSI	SNR
-----------	---------	----	-----------	------	-----

**Tableau 0.1** : Trame des données LoRA.

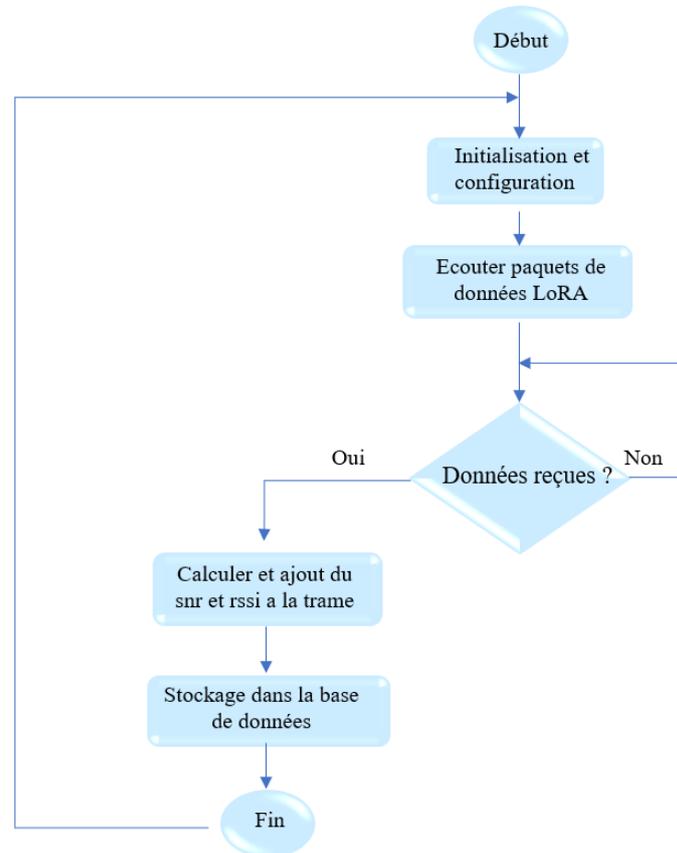
Le Raspberry Pi réceptionne cette trame, estime les valeurs de RSSI et de SNR, puis les enregistre dans la base de données locale précédemment traitement de données.

Les organigrammes (figure 3.9 et figure 3.10) illustrent la logique de fonctionnement de notre solution.



**Figure 0.9** : Organigramme de nœud mobile sous ESP32.

Le nœud mobile commence par initialiser et configurer ses paramètres de communication, puis tente d'établir une connexion avec la passerelle. En cas d'échec, il réitère la connexion jusqu'à ce qu'elle soit réussie, assurant ainsi que les données (identifiant, mesures, etc.) ne soient transmises que lorsque la liaison est stable.



**Figure 0.10** : Organigramme de passerelle sous Raspberry Pi.

La passerelle, quant à elle, initialise ses paramètres et reste en écoute continue des paquets LoRa. Lorsqu'un paquet est reçu, elle calcule les indicateurs de qualité du signal, tels que le RSSI et le SNR, qu'elle ajoute à la trame avant de stocker les données complètes dans une base pour exploitation ultérieure.

Ce processus assure la fiabilité des échanges en garantissant que les transmissions ne sont effectuées qu'en présence d'une liaison stable, tout en permettant à la passerelle de collecter et d'enrichir les informations reçues pour un traitement optimal.

### 3.3 Traitement De Données

Les valeurs RSSI enregistrées dans la base de données sont d'abord extraites, puis soumises à une phase de segmentation (Fenêtrage). À partir de chaque segment, un ensemble de caractéristiques statistiques est calculé afin d'alimenter un modèle d'apprentissage automatique destiné à estimer la position.

Pour évaluer les performances des modèles développés, plusieurs métriques ont été employées. Celles-ci permettent non seulement de juger de la précision globale du modèle, mais également d'analyser son comportement spécifique pour chaque classe de position. Les indicateurs retenus sont détaillés ci-dessous [66] :

- **Exactitude (Accuracy)** : Représente la proportion totale de prédictions correctes. Bien qu'intuitive, cette métrique peut s'avérer peu représentative en présence d'un déséquilibre entre les classes.

$$\text{Exactitude} = \frac{VP + VN}{VP + VN + FP + FN} \quad \dots (3.1)$$

- **Précision (Precision)** : Évalue la capacité du modèle à limiter les faux positifs. Elle est particulièrement utile lorsque l'enjeu principal est de minimiser les classifications incorrectes.

$$\text{Précision} = \frac{VP}{VP + FP} \quad \dots (3.2)$$

- **Rappel (Recall)** : Mesure l'aptitude du modèle à identifier toutes les instances d'une classe donnée, en réduisant les faux négatifs. Cette métrique est essentielle dans un contexte de localisation où l'omission d'une position peut être critique.

$$\text{Rappel} = \frac{VP}{VP + FN} \quad \dots (3.3)$$

- **F1-Score** : Combine de manière harmonique la précision et le rappel, fournissant ainsi une évaluation équilibrée, notamment en cas de classes déséquilibrées ou lorsque les coûts d'erreur sont similaires.

$$F1 = 2 \cdot \frac{\textit{Précision} \cdot \textit{rappel}}{\textit{Précision} + \textit{rappel}} \quad \dots (3.4)$$

- VP (Vrai Positif) : cas correctement classés comme positifs,
- VN (Vrai Négatif) : cas correctement classés comme négatifs,
- FP (Faux Positif) : cas incorrectement classés comme positifs,
- FN (Faux Négatif) : cas incorrectement classés comme négatifs.

- **Courbe ROC et AUC (Area Under Curve) :**

La courbe ROC (Receiver Operating Characteristic) permet d'évaluer la performance d'un modèle de classification en mesurant sa capacité à distinguer les classes. Elle trace le taux de vrais positifs (TVP) en fonction du taux de faux positifs (TFP) pour différents seuils de décision :

- TVP (taux de vrais positifs) :

$$TVP(\textit{rappel}) = \frac{VP}{VP + FN} \quad \dots (3.5)$$

- TFP (taux de faux positifs) :

$$TFP(\textit{rappel}) = \frac{FP}{FP + VN} \quad \dots (3.6)$$

La surface sous la courbe ROC, appelée AUC (Area Under the Curve), donne une mesure globale de la performance :

- **AUC = 1** : classification parfaite
- **AUC=0,5** : performance équivalente au hasard
- **AUC < 0,5** : mauvaise performance (pire que le hasard)

Dans le cas d'un problème multi-classe, on utilise une approche "un contre tous" (one-vs-rest) : on trace une courbe ROC pour chaque classe en la comparant aux autres [66]. Cela permet d'évaluer la capacité du modèle à identifier correctement chaque classe séparément.

### **3.4 Conclusion**

Ce chapitre présente en profondeur la conception de notre système basé sur la technologie LoRa. L'objectif était de créer une solution économique et efficace, capable de collecter et de transmettre des données sur de longues distances, tout en garantissant une connexion fiable et optimisée. Les choix technologiques et les configurations sélectionnées établissent ainsi les bases pour la mise en œuvre et l'évaluation des performances du système dans le cadre de notre recherche.

**CHAPITRE -4 :**

**MESURES, TRAITEMENT ET RESULTATS**

## 4.1 Introduction

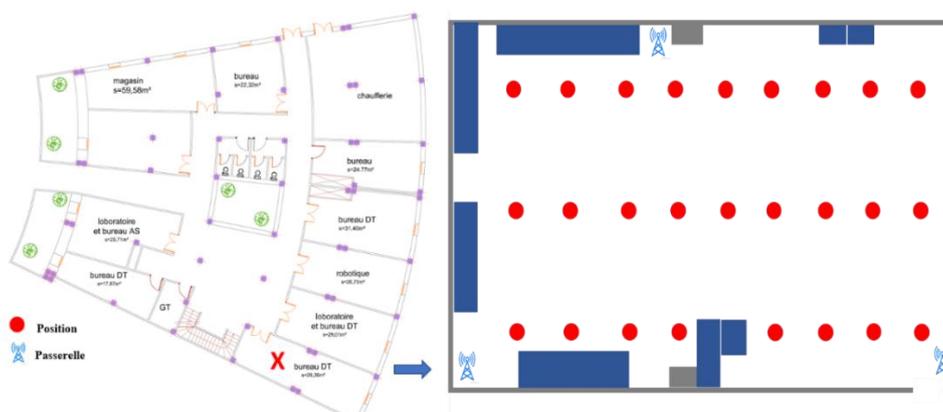
Dans le chapitre précédent, nous avons présenté en détail l'implémentation matérielle et logicielle du système de localisation proposé, en décrivant les composants utilisés, leur configuration ainsi que l'architecture générale du dispositif. Dans le présent chapitre, nous nous intéressons à l'évaluation expérimentale des performances de ce système dans différents environnements intérieurs. L'objectif est de mesurer la précision de la localisation en fonction de plusieurs paramètres, tels que le nombre et la position des passerelles LoRa, la structure spatiale du site de test, ainsi que les méthodes de traitement et de classification des données RSSI.

## 4.2 Compagnes De Mesure

Des campagnes de mesure ont été conduites dans la Division Telecom-CDTA, dans des environnements variés, allant de simples bureaux jusqu'à plusieurs niveaux d'un bâtiment.

- **Scénario 1 – Expérimentation dans un Bureau**

Pour ce premier scénario, les mesures RSSI ont été collectées dans un bureau de 10 x 5 mètres comportant divers obstacles comme des murs et du mobilier. Trois passerelles (Gateways) ont été installées à des emplacements fixes prédéfinis, tandis que le nœud émetteur a été déplacé selon une grille de 27 positions. Cette configuration vise à analyser l'impact des éléments obstructifs sur la précision d'une localisation fondée sur les données RSSI. L'environnement expérimental est illustré à la figure 4.1.



**Figure 0.1:** Environnement expérimental du Scénario 1.

- **Scénario 2 – Expérimentation dans un couloir**

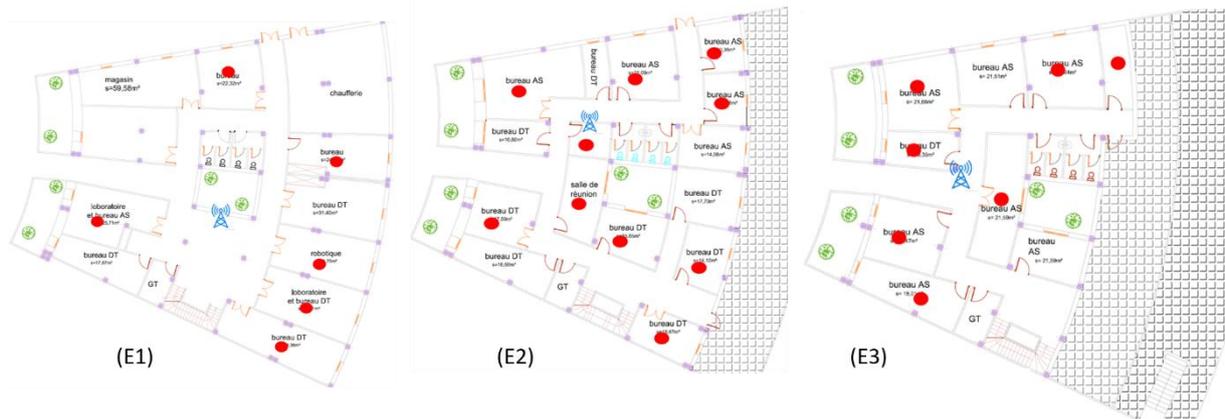
Dans ce scénario, les tests ont été élargis à un couloir intérieur où 3 Gateways ont été placées : deux dans des pièces adjacentes, une dans un espace central fermé. L'émetteur a été déplacé tous les 2 m le long du couloir, avec 33 points mesurés. Ce scénario simule une zone étendue avec obstacles intermédiaires. La figure 4.2. Présente l'environnement de l'expérimentation.



**Figure 0.2 :** Environnement expérimental du Scénario 2.

- **Scénario 3 – Configuration multi-étages**

Trois passerelles ont été installées à des étages différents du bâtiment. Les mesures ont été effectuées dans plusieurs bureaux répartis sur chaque étage : 6 au rez-de-chaussée (E1), 10 au premier étage (E2), et 7 au second (E3). L'objectif principal est d'observer la capacité du système à discriminer des positions verticalement séparées. La figure 4.3. Illustre l'environnement de travail.



**Figure 0.3 :** Environnement expérimental du Scénario 3.

### 4.3 Acquisition Des Données

Pour chaque position de mesure, environ 800 échantillons de puissance du signal reçu (RSSI) ont été collectés à l'aide de modules LoRa opérant à une fréquence de 433 MHz. Ces modules ont été configurés avec des paramètres fixes : une puissance d'émission de 13 dBm, un facteur d'étalement (Spreading Factor) de 7, un taux de codage de 4/5, ainsi qu'une bande passante de 125 kHz. Les dispositifs ont été positionnés à une hauteur d'un mètre à l'aide de balises prévues à cet effet. Le maintien constant de ces paramètres visait à garantir la validité des comparaisons entre les différentes configurations expérimentales. La figure 4.4 présente des photographies illustrant le dispositif mis en place.



**Figure 0.4 :** Images des expérimentations.

Le processus de collecte des données a été entièrement automatisé à l'aide d'un script Python déployé sur un Raspberry Pi faisant office de passerelle. Les mesures sont systématiquement enregistrées dans une base de données, accompagnées de métadonnées pertinentes telles que l'horodatage, l'identifiant de la passerelle et l'étiquette correspondant à la position de mesure. La figure 4.5 présente une vue d'ensemble de la collecte et des données reçues

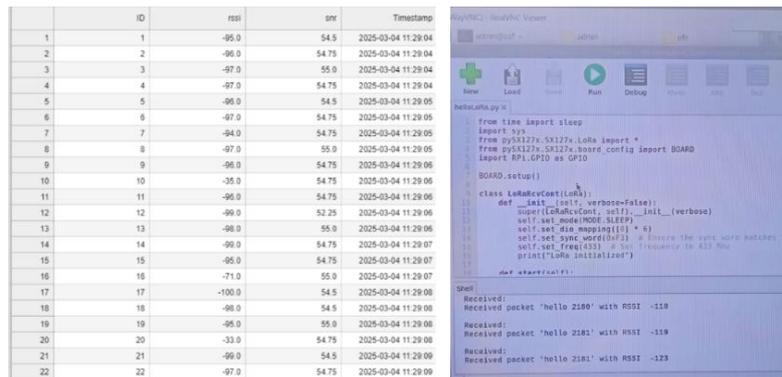


Figure 0.5 : Collecte des données.

#### 4.4 Évaluation Des Performances

L'ensemble du jeu de données a été divisé en deux sous-ensembles :

- 80 % ont été utilisés pour l'entraînement des modèles (K-Nearest Neighbors, Support Vector Machine et Random Forest),
- 20 % restants ont été réservés pour l'évaluation des performances sur des données non vues.

Dans le cadre de l'extraction des caractéristiques pour l'analyse des signaux LoRa, deux approches de fenêtrage ont été considérées :

- Le fenêtrage par nombre de données (samples)
- Le fenêtrage par unité de temps.

Le premier consiste à regrouper un nombre fixe d'échantillons (par exemple, 10 lignes), ce qui simplifie le traitement lorsque le taux de réception est régulier. La seconde repose sur des fenêtres temporelles fixes (ex. toutes les 2 secondes), permettant une meilleure synchronisation temporelle, notamment en cas de variation du taux d'échantillonnage.

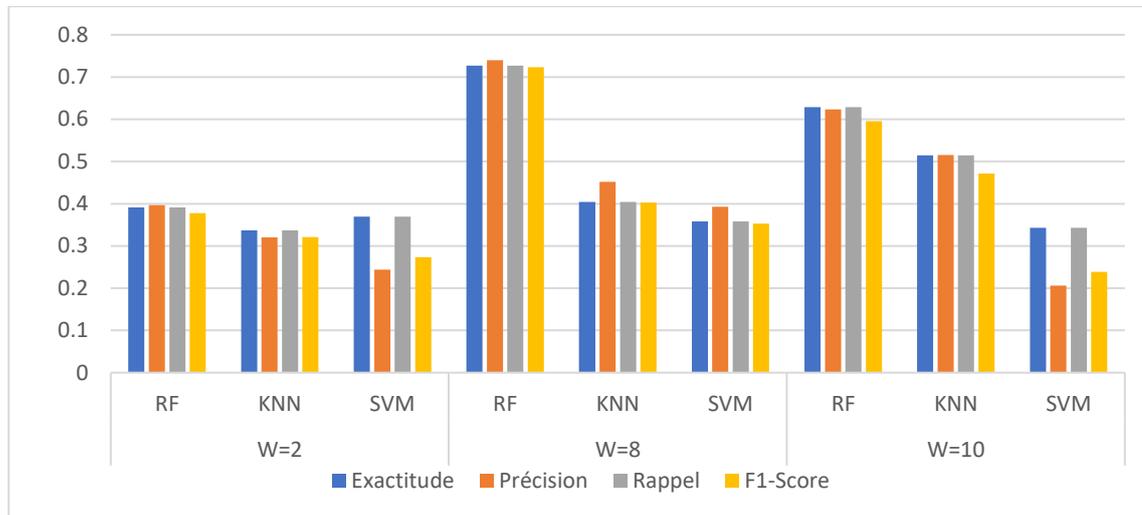
Dans ce projet, le fenêtrage par nombre de données a été privilégié pour sa simplicité, sa cohérence avec la structure des données collectées, et sa capacité à capturer l'évolution locale

du signal tout en réduisant le bruit. Pour chaque segment, quatre caractéristiques statistiques ont été extraites, comme présenté dans le tableau 4.1. Ces features servent ensuite d'entrée aux modèles d'apprentissage automatique.

Caractéristique	Impact potentiel	Equation
<b>Moyenne Mean</b>	Représente la puissance moyenne du signal LoRa sur la période	$RSSI_{Moy} = \frac{1}{n} \sum_{i=0}^n RSSI_i$ <ul style="list-style-type: none"> <li>• <math>\frac{1}{n}</math> : Division par le nombre total de valeurs (la moyenne).</li> <li>• <math>RSSI_{Moy}</math> : Rssi moyen</li> </ul>
<b>L'écart-type Std</b>	Reflète les variations rapides du signal, souvent causées par des obstacles, des mouvements, ou la multipath.	$STD_{RSSI} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (RSSI_i - RSSI_{Moy})^2}$ <ul style="list-style-type: none"> <li>• <math>STD_{RSSI}</math> : écart-type des valeurs de RSSI.</li> <li>• <math>RSSI_i</math> : i-ème valeur de RSSI.</li> <li>• <math>RSSI_{Moy}</math>: moyenne des valeurs RSSI.</li> <li>• <math>n</math> : nombre total de mesures.</li> </ul>
<b>Le minimum et le maximum Min / max</b>	Donnent des bornes extrêmes du RSSI dans l'intervalle.	$RSSI_{min} = \min(RSSI_1, RSSI_2, \dots, RSSI_n)$ $RSSI_{max} = \max(RSSI_1, RSSI_2, \dots, RSSI_n)$ <ul style="list-style-type: none"> <li>• <math>RSSI_i</math> : i-ème valeur de RSSI.</li> <li>• <math>n</math> : nombre total de mesures.</li> <li>• <math>min</math>: fonction qui renvoie la plus petite valeur</li> <li>• <math>max</math> : fonction qui renvoie la plus grande valeur.</li> <li>• <math>RSSI_{min}</math>: plus faible puissance</li> <li>• <math>RSSI_{max}</math> : meilleure puissance</li> </ul>

**Tableau 0.1** : Caractéristiques temporelles.

L'influence de la taille de la fenêtre a également été étudiée en comparant les performances des modèles pour différentes valeurs de  $W$  (2, 8 et 10) comme ce présente dans la figure 4.6 :



**Figure 0.6 :** Performances du scénario 1 pour différentes tailles de segment.

Les résultats ont montré que la fenêtre de taille  $W = 8$  offrait les meilleures performances globales, en particulier avec le modèle Random Forest. Ces performances traduisent un bon compromis entre richesse de l'information et stabilité des données, justifiant ainsi le choix de cette configuration pour la suite des analyses.

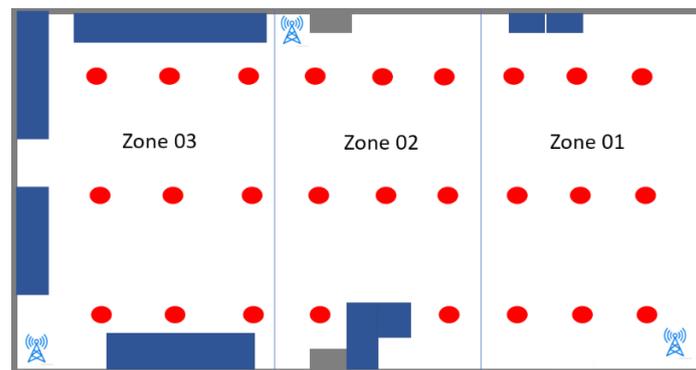
#### 4.4.1 Scénario 1

Dans le scénario 1, 27 positions distinctes ont été considérées, avec l'utilisation de trois passerelles LoRa déployées à des emplacements fixes. Trois stratégies d'apprentissage ont été évaluées :

- **Stratégie 1\_ Modèles mono-passerelle (individuels) :** Dans cette configuration, un modèle est entraîné séparément pour chacune des passerelles, en utilisant uniquement ses propres données RSSI. Cette stratégie permet d'évaluer la capacité de chaque passerelle, prise isolément, à assurer une localisation fiable, et de comparer ces performances à celles obtenues avec une approche fusionnée.
- **Stratégie 2 \_ Modèle multi-passerelles (fusion globale) :** Cette approche consiste à regrouper les mesures RSSI provenant des trois passerelles dans un seul ensemble de données. Un modèle unique est ensuite entraîné à partir de ces données combinées.

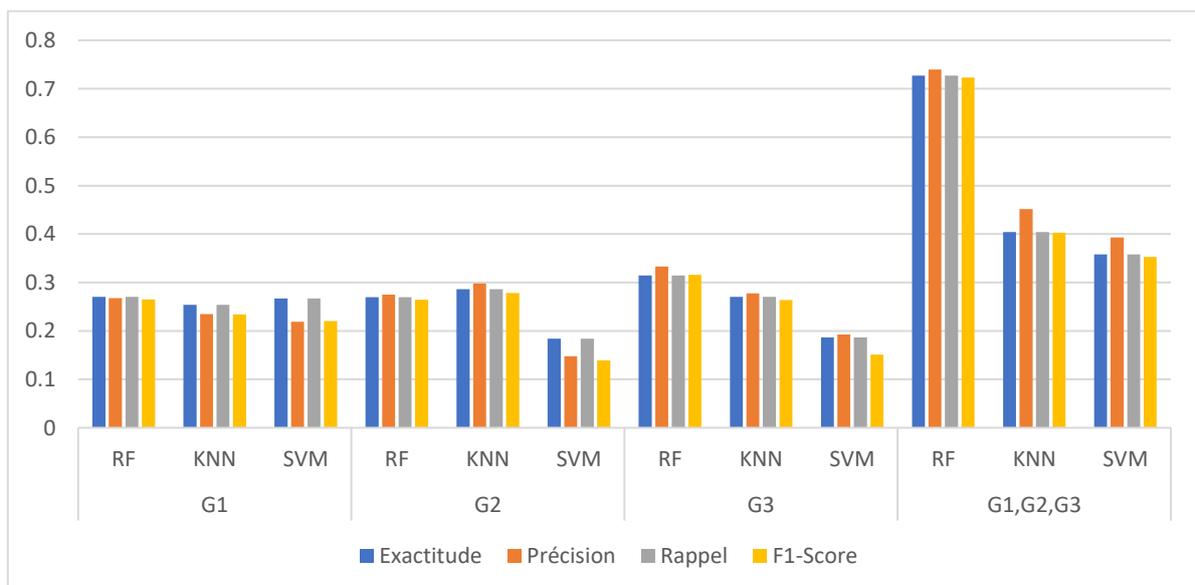
L'objectif est d'exploiter la diversité spatiale offerte par les différentes passerelles pour améliorer la précision globale de la localisation.

- **Stratégie 3\_Classification zonale** : Les 27 positions ont été regroupées en trois zones en fonction de leur proximité spatiale, comme illustré à la figure 4.7. Cette stratégie consiste à reformuler la tâche de classification en une problématique à trois classes correspondant aux zones, plutôt qu'à 27 positions individuelles. Cette simplification permet d'évaluer la robustesse du système dans un contexte où une localisation de niveau zone est suffisante.



**Figure 0.7** : Division en zone\_S1.

Les performances associées à chaque stratégie ont été représentées à l'aide de courbes ROC et d'histogrammes. Les figures 4.8 et 4.9 présentent les résultats obtenus pour les stratégies 1 et 2, tandis que les figures 4.10 et 4.11 illustrent ceux relatifs à la stratégie 3.



**Figure 0.8** : Performances selon les stratégies 1 et 2\_S1.

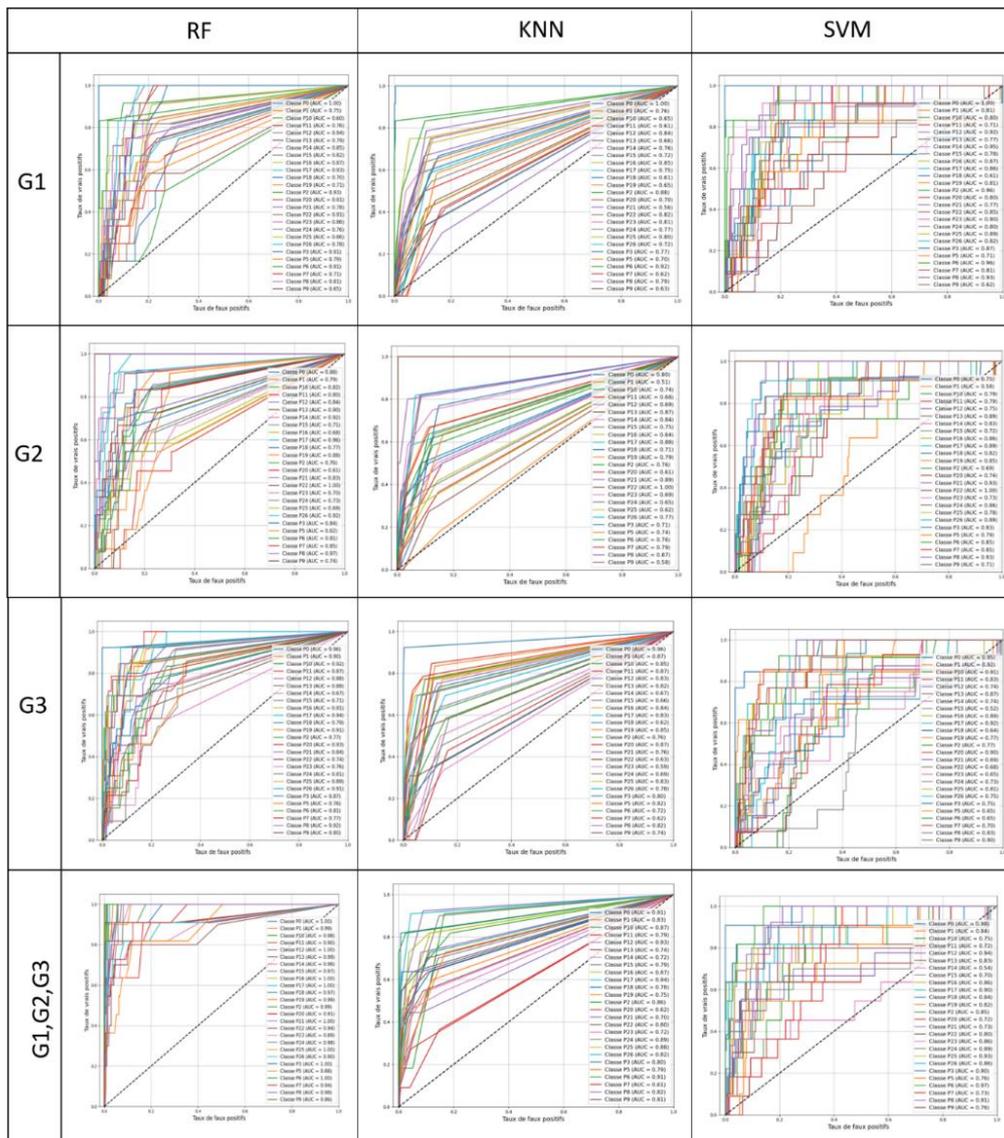


Figure 0.9 : Courbes ROC des stratégies 1 et 2\_S1.

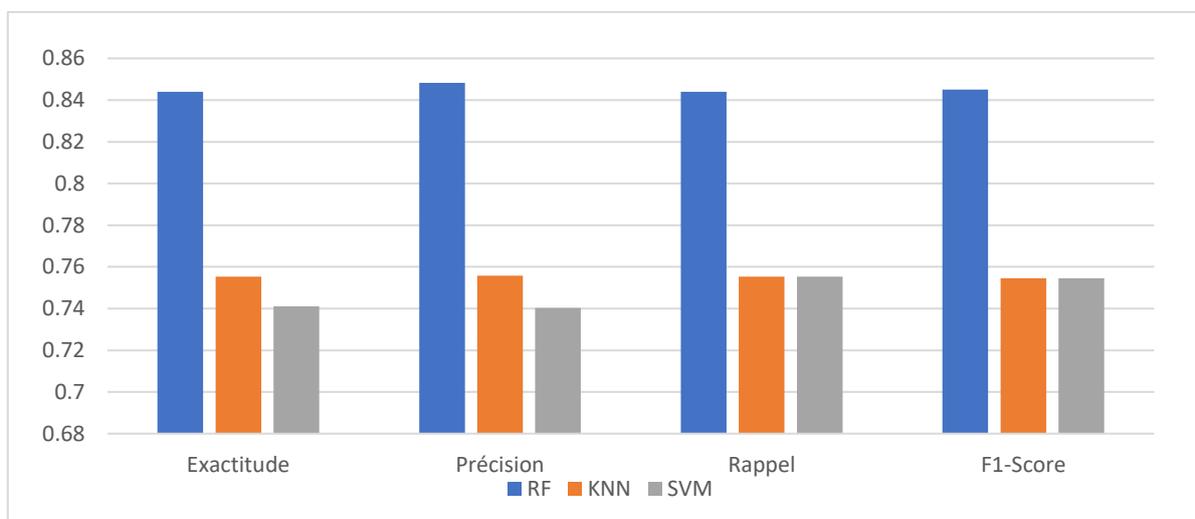
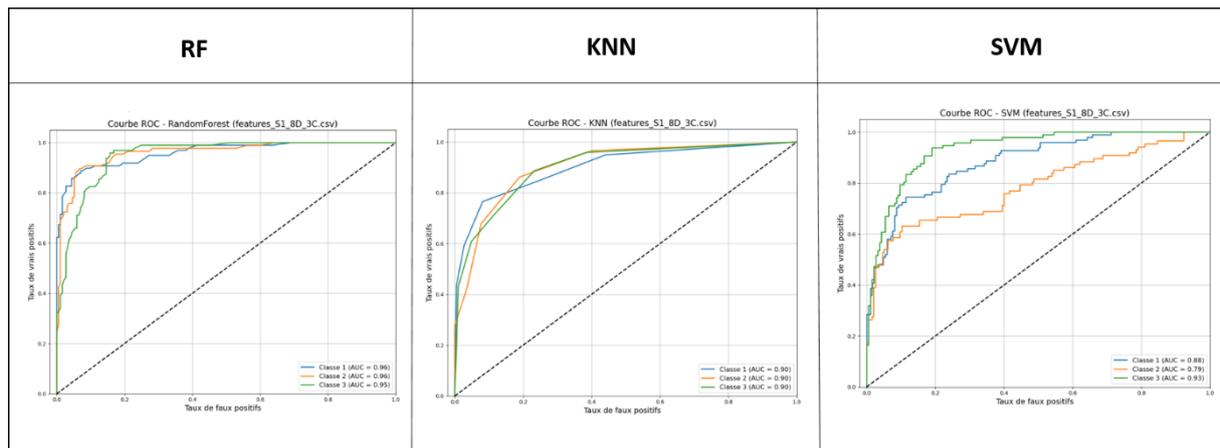


Figure 0.10 : Performance selon la stratégie 3\_S1.



**Figure 0.11** : Courbes ROC de la stratégie 3\_S1.

L'analyse des performances montre que l'entraînement avec une seule passerelle fournit des résultats globalement limités, quel que soit l'algorithme utilisé. Les performances varient selon l'emplacement des passerelles, mais restent faibles dans l'ensemble. Parmi les algorithmes testés, Random Forest dépasse légèrement KNN et SVM, ce dernier affichant les résultats les plus faibles, ce qui suggère une moindre adéquation à ce type de données.

Une forte confusion entre les positions et une grande variabilité des courbes ROC indiquent que les mesures RSSI issues d'une seule passerelle ne suffisent pas à discriminer efficacement les classes. En revanche, l'association des données des trois passerelles améliore sensiblement les performances de tous les algorithmes, en particulier pour Random Forest, qui se démarque avec une meilleure précision de classification et des AUC plus élevés. KNN progresse également mais reste en retrait par rapport à Random Forest, tandis que SVM montre les améliorations les plus faibles, conservant une capacité de discrimination limitée.

La réduction du problème à trois classes améliore encore les résultats, en rendant la classification plus robuste aux variations du signal. Les performances dans ce cadre indiquent une meilleure fiabilité de la localisation par zones que celle fondée sur des positions individuelles.

Ainsi, l'intégration multi-passerelles combinée à une classification par zones apparaît comme la stratégie la plus efficace, avec Random Forest comme algorithme le plus performant dans ce contexte.

#### 4.4.2 Scénario 2

Dans ce scénario, les résultats ont été présentés uniquement à l'aide de l'algorithme Random Forest, sélectionné pour ses performances globalement supérieures dans les scénarios précédents. Concernant la classification des 33 positions, seule la stratégie du modèle global, reposant sur la combinaison des données des trois passerelles, a été retenue pour l'analyse. En revanche, dans le cadre de la classification par zones, où le couloir a été subdivisé en quatre zones distinctes, comme illustré à la figure 4.12.

Deux approches ont été comparées : d'une part, l'entraînement d'un modèle global à partir des données fusionnées des trois passerelles, et d'autre part, des modèles indépendants construits à partir des données propres à chaque passerelle.



Figure 0.12 : Division en zone\_S2.

Les résultats obtenus sont représentés dans les figures 4.13 et 4.14.

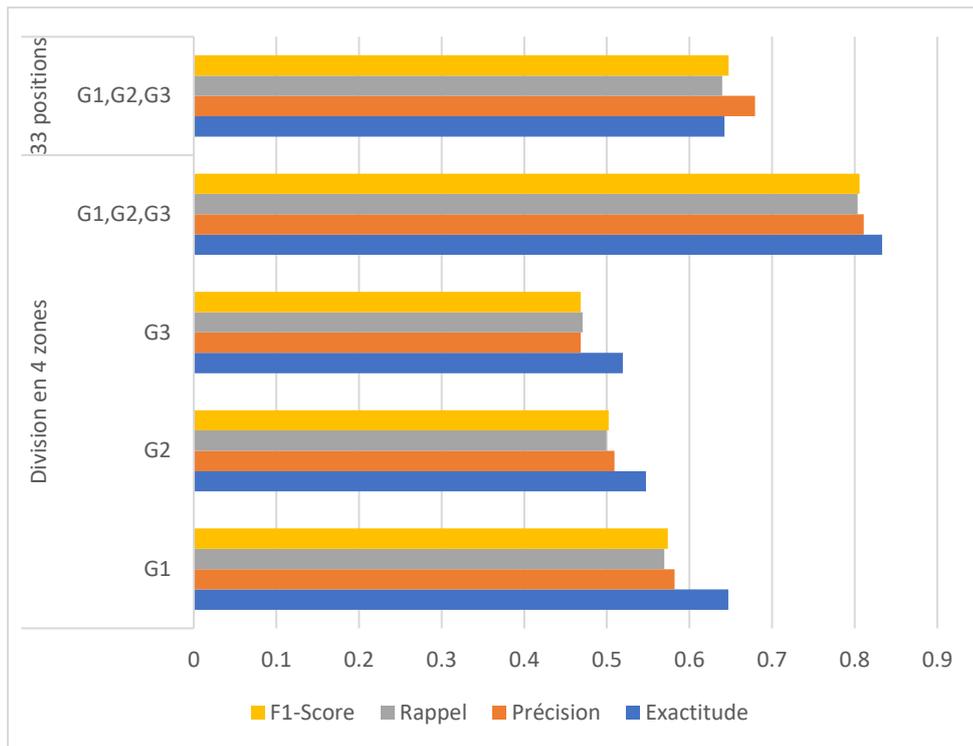


Figure 0.13 : Performances selon les stratégies 1 et 2\_S2.

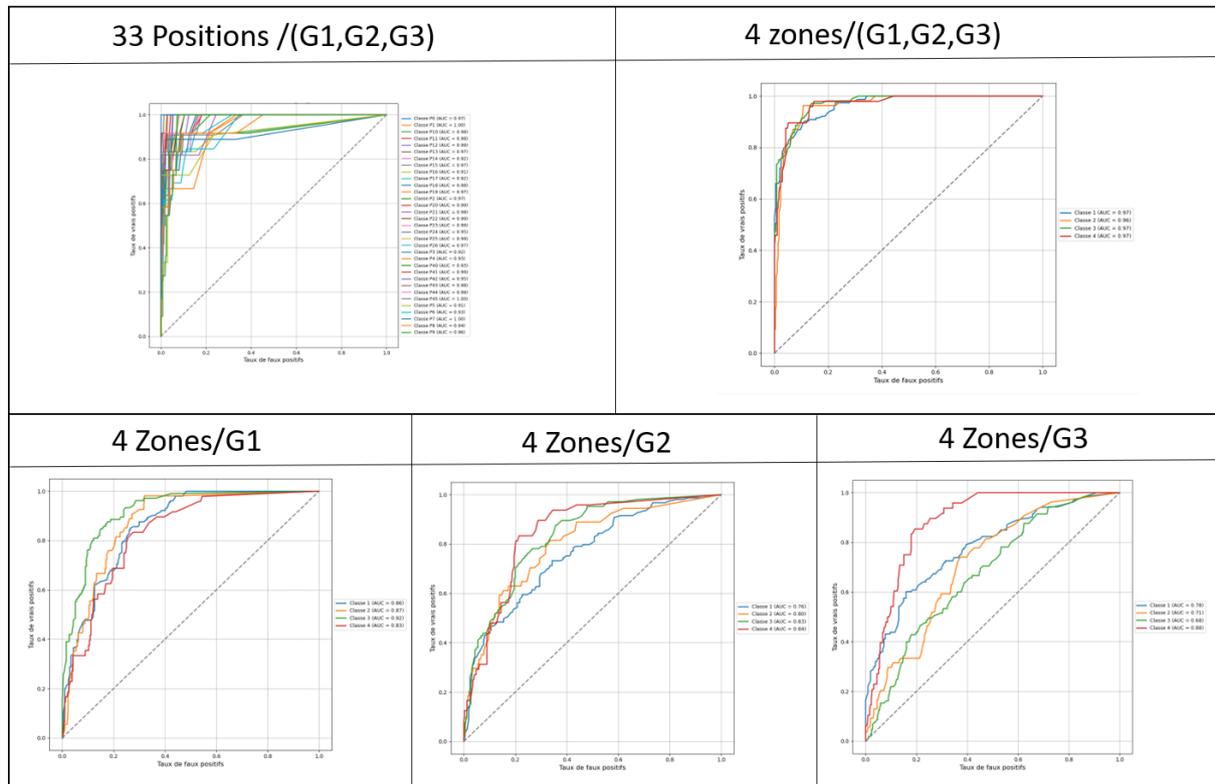


Figure 0.14 : Courbes ROC\_S2.

Les résultats obtenus montrent une performance modérée du modèle pour la classification des 33 positions, avec une exactitude de 64,2 %, un F1-score de 64,7 % et des valeurs de précision et de rappel similaires, traduisant une capacité partielle à distinguer correctement l'ensemble des positions. Les courbes ROC révèlent des AUC généralement élevées, certaines atteignant 1,0, indiquant une bonne différenciation pour plusieurs classes. Lors de l'évaluation sur 4 classes à l'aide de trois passerelles indépendantes, les performances varient légèrement selon la passerelle, ce qui suggère une influence de leur positionnement sur la qualité de la classification. Les valeurs d'AUC, comprises entre 0,68 et 0,92, indiquent une capacité modérée à séparer les classes dans ce contexte. En revanche, lorsque les données des trois passerelles sont combinées, les résultats s'améliorent nettement, confirmant l'intérêt de cette approche pour renforcer la précision de localisation. Les courbes ROC correspondantes, proches du coin supérieur gauche, traduisent une excellente capacité de discrimination entre les quatre zones considérées.

#### 4.4.3 Scénario 3

Dans le scénario 3, deux stratégies de classification ont été mises en œuvre afin d'exploiter les mesures RSSI collectées sur 23 positions réparties sur trois étages, chaque étage étant équipé d'une passerelle dédiée.

- **Stratégie 1** : une classification directe a été effectuée en considérant l'ensemble des 23 positions comme des classes distinctes, en s'appuyant sur les données issues des trois passerelles positionnées respectivement à chaque niveau du bâtiment.
- **Stratégie 2** : une approche hiérarchique a été adoptée. Elle consiste, dans un premier temps, à identifier l'étage à partir des données concaténées de chaque niveau. Une fois l'étage reconnu, une classification locale est effectuée pour déterminer le bureau au sein de l'étage correspondant.

Les résultats obtenus sont représentés dans les figures 4.15 et 4.16.

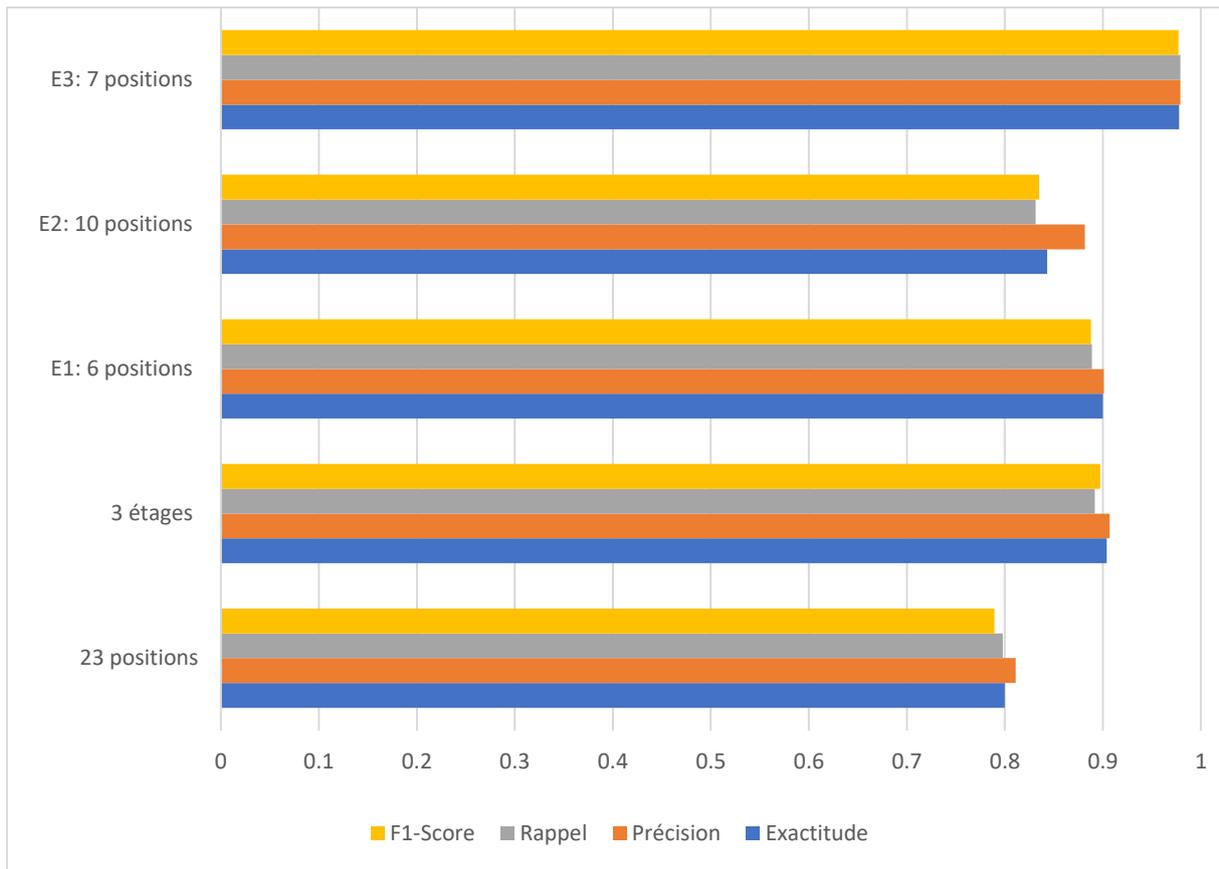


Figure 0.15 : Performances selon les stratégies 1 et 2\_S3.

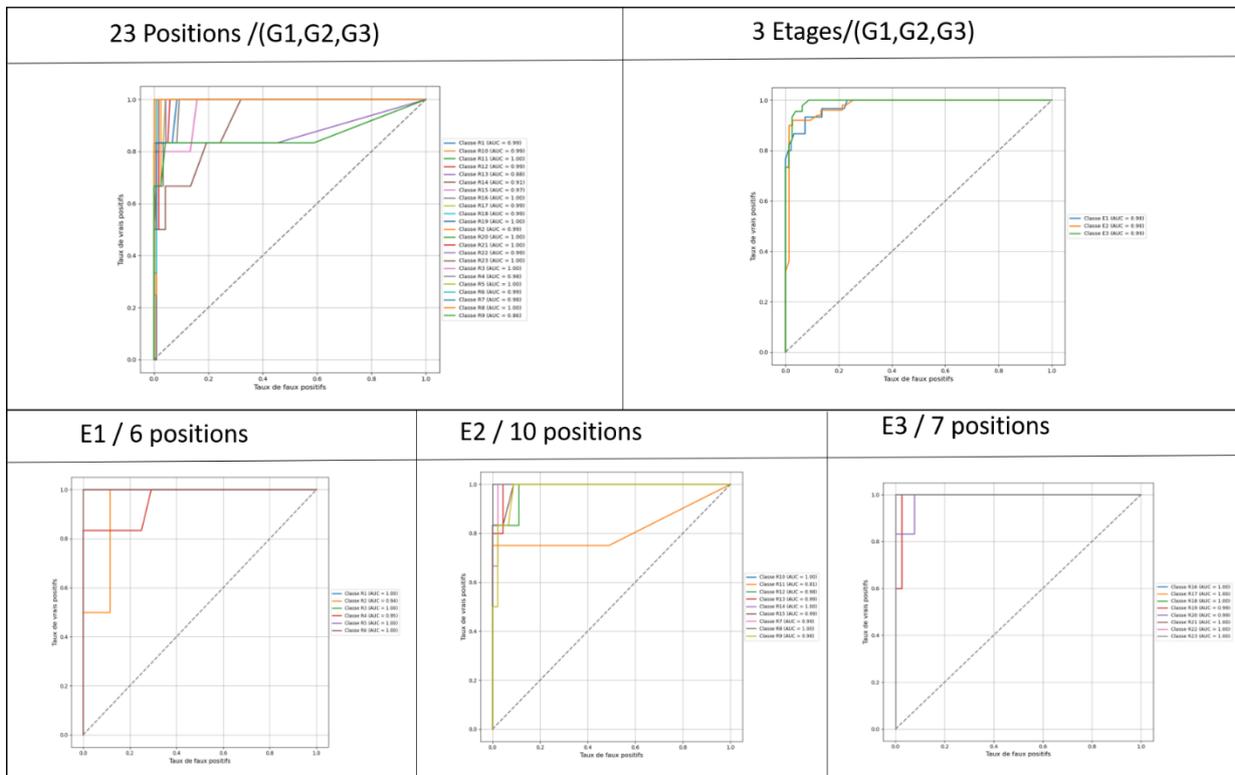


Figure 0.16 : Courbes ROC\_S3.

Les résultats quantitatifs obtenus confirment la pertinence de l'approche hiérarchique pour la localisation en environnement multi-étage. Lorsqu'un modèle unique est entraîné pour distinguer directement les 23 positions, une exactitude de 0,80 est obtenue, avec un F1-Score de 0,79.

En revanche, l'adoption d'une classification hiérarchique ; consistant à identifier dans un premier temps l'étage (3 classes), puis à localiser précisément la position dans l'étage ; permet une amélioration significative des performances. La classification des étages seule atteint une exactitude de 0,90 et un F1-Score de 0,89.

Pour la localisation intra-étage, les résultats varient selon les niveaux : l'étage 3 affiche les meilleures performances avec une exactitude de 0,98 et un F1-Score de 0,98, suivi de l'étage 1 (F1-Score de 0,89) et de l'étage 2, qui présente des performances légèrement inférieures (F1-Score de 0,83).

Ces résultats montrent clairement que la décomposition du problème en deux étapes permet non seulement de réduire la complexité de la tâche, mais également d'augmenter la précision globale du système, en particulier lorsque les conditions de mesure sont plus homogènes au sein d'un même étage.

#### **4.5 Discussion Des Résultats**

L'ensemble des résultats expérimentaux obtenus à travers les trois scénarios met en évidence plusieurs tendances significatives concernant la précision du système de localisation basé sur les mesures RSSI en environnement intérieur. Tout d'abord, l'influence du nombre de passerelles utilisées est clairement démontrée : dans tous les cas, l'utilisation de plusieurs passerelles, notamment leur combinaison dans un modèle global, conduit à une amélioration substantielle des performances de classification. Cette observation est constante, tant pour la localisation fine (positions individuelles) que pour la classification spatiale par zones.

Dans le scénario 1, les résultats montrent que les modèles mono-passerelle, entraînés séparément, présentent des performances limitées, avec des différences notables entre passerelles selon leur position dans l'espace. En revanche, la fusion des données des trois passerelles améliore sensiblement la capacité du système à discriminer les positions.

De plus, la classification par zones (réduction du problème à trois classes) permet d'accroître encore la robustesse du système, en particulier lorsque les mesures RSSI sont soumises à des perturbations dues à l'environnement.

Le scénario 2, appliqué à un environnement de couloir, confirme cette tendance. La classification directe des 33 positions reste modérée malgré l'usage d'un modèle global, mais l'approche par zones, en divisant l'espace en quatre classes, montre une nette amélioration des indicateurs de performance. Là encore, la combinaison des données issues des trois passerelles offre de meilleurs résultats que les modèles entraînés individuellement, avec des courbes ROC indiquant une capacité de discrimination renforcée entre les zones.

Dans le scénario 3, impliquant une configuration multi-étage, la comparaison entre une classification directe des 23 positions et une approche hiérarchique s'avère particulièrement instructive. La stratégie hiérarchique, qui consiste d'abord à identifier l'étage, puis à localiser la position à l'intérieur de celui-ci, permet d'obtenir une nette amélioration des résultats. Les performances par étage montrent des variations : l'étage 3 présente les meilleurs résultats, tandis que l'étage 2 affiche des scores légèrement inférieurs. Cette structuration en deux étapes apparaît donc comme une méthode efficace pour traiter la complexité croissante des environnements verticaux, en réduisant l'espace de décision à chaque étape.

Dans l'ensemble, l'algorithme Random Forest s'est distingué par ses performances globalement supérieures à celles de KNN et SVM dans l'ensemble des scénarios testés. Sa robustesse face aux variations du signal RSSI, souvent bruité et instable en intérieur, en fait un choix particulièrement adapté pour ce type d'application.

Les résultats obtenus confirment l'intérêt d'une stratégie combinée : fusion des données multi-passerelles, segmentation spatiale par zones ou étages, et utilisation d'algorithmes adaptés comme Random Forest pour garantir une localisation plus fiable et précise.

## **4.6 Conclusion**

Les expérimentations réalisées dans ce chapitre ont permis d'évaluer concrètement les capacités du système de localisation LoRa dans divers environnements intérieurs. Les résultats montrent que la précision du système dépend fortement de la stratégie de déploiement des passerelles, de la structure spatiale de l'environnement et de la méthode de classification utilisée. L'utilisation d'un modèle global intégrant les données de plusieurs passerelles s'est révélée plus efficace que les approches mono-passerelle, en particulier pour des espaces complexes.

La classification par zones a permis d'améliorer la robustesse du système en réduisant la granularité du problème, tandis que la configuration hiérarchique, appliquée aux environnements multi-étages, a significativement renforcé la précision en segmentant la tâche en deux étapes successives (étage puis position). L'algorithme Random Forest a démontré une supériorité constante dans tous les cas de figure.

Ces résultats valident l'approche adoptée et ouvrent la voie à une intégration du système dans des applications réelles nécessitant une localisation fiable en intérieur.

## CONCLUSION GENERALE

Ce travail a porté sur la conception, le développement et l'évaluation expérimentale d'un système de localisation en intérieur basé sur la technologie LoRa, couplée à des algorithmes d'apprentissage automatique. Trois environnements représentatifs ont été explorés : un bureau, un couloir, et un bâtiment à plusieurs étages. Les données RSSI collectées via trois passerelles ont été prétraitées (nettoyage, synchronisation, étiquetage), puis utilisées pour entraîner plusieurs modèles de classification supervisée, notamment KNN, SVM et Random Forest.

Les résultats obtenus ont permis de comparer les performances de ces algorithmes dans différents contextes. Random Forest s'est distingué par sa robustesse et sa capacité à fournir les meilleurs résultats dans la majorité des scénarios testés. Ces travaux contribuent à démontrer la faisabilité technique de la localisation indoor avec LoRa, tout en mettant à disposition un jeu de données expérimental réutilisable pour des recherches futures.

Néanmoins, certaines limites ont été identifiées : la variabilité des signaux RSSI, le déséquilibre des données entre passerelles, le nombre limité de points de mesure, ainsi que l'absence de synchronisation temporelle fine. Ces éléments ont pu impacter la précision du système, notamment dans les environnements verticaux plus complexes.

Plusieurs axes d'amélioration sont envisageables : augmentation du nombre de passerelles, exploitation de nouvelles caractéristiques du signal, hybridation avec d'autres technologies de positionnement, ou encore recours à des modèles plus avancés comme le deep learning. Par ailleurs, des perspectives innovantes peuvent être explorées, telles que l'intégration de systèmes auto-adaptatifs, la localisation collaborative entre passerelles, l'utilisation de données synthétiques générées par IA, ou l'embarquement de modèles via TinyML pour des dispositifs légers et autonomes. Enfin, l'intégration dans une architecture Smart Building ouvre la voie à des applications contextuelles avancées et automatisées.

Ainsi, ce mémoire pose les bases d'une solution de localisation intérieure évolutive, économe en énergie et adaptable à des environnements complexes, avec un fort potentiel de déploiement dans les contextes industriels, hospitaliers ou domotiques.

## BIBLIOGRAPHIES

- [1] K. Gherib, « GEOLOCALISATION DES PERSONNES EN ENVIRONNEMENT INDOOR VIA LA FUSION DES CAPTEURS DE SMARTPHONE », Thesis, UNIVERSITÉ KASDI MERBAH OUARGLA, 2021. Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <http://dspace.univ-ouargla.dz/jspui/handle/123456789/29135>
- [2] J. Kostak, « Indoor GPS Tracking », Sewio RTLS. Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://www.sewio.net/indoor-gps-tracking/>
- [3] T. G. Hailu, X. Guo, H. Si, L. Li, et Y. Zhang, « Theories and Methods for Indoor Positioning Systems: A Comparative Analysis, Challenges, and Prospective Measures », *Sensors*, vol. 24, n° 21, Art. n° 21, janv. 2024, doi: 10.3390/s24216876.
- [4] F. Evennou, « Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor », phdthesis, Université Joseph-Fourier - Grenoble I, 2007. Consulté le: 18 mai 2025. [En ligne]. Disponible sur: <https://theses.hal.science/tel-00136064>
- [5] O. G., « A Survey of ZigBee Wireless Sensor Network Technology: Topology, Applications and Challenges », *Int. J. Comput. Appl.*, vol. 130, n° 9, p. 47-55, nov. 2015, doi: 10.5120/ijca2015907130.
- [6] E. Hatem, « Optimized And Sub-Metric Indoor Localization System Based On Uhf Rfid Technology », phdthesis, Université Paris-Est ; Université Libanaise, 2021. Consulté le: 18 mai 2025. [En ligne]. Disponible sur: <https://theses.hal.science/tel-03639890>
- [7] M. Anjum, M. Abdullah Khan, S. A. Hassan, H. Jung, et K. Dev, « Analysis of time-weighted LoRa-based positioning using machine learning », *Comput. Commun.*, vol. 193, p. 266-278, sept. 2022, doi: 10.1016/j.comcom.2022.07.010.
- [8] B. Ozdenizci, V. Coskun, et K. Ok, « NFC Internal: An Indoor Navigation System », *Sensors*, vol. 15, n° 4, Art. n° 4, avr. 2015, doi: 10.3390/s150407571.
- [9] L. Bencharif, « Localisation en intérieur basée sur l'apprentissage automatique : exploitation des informations sur l'état des canaux Wi-Fi », masters, Université du Québec à Trois-Rivières, Trois-Rivières, 2020. Consulté le: 18 mai 2025. [En ligne]. Disponible sur: <https://depot-e.uqtr.ca/id/eprint/9546/>
- [10] A. Ramamonjy, « Développement de nouvelles méthodes de classification/localisation de signaux acoustiques appliquées aux véhicules aériens », phdthesis, Conservatoire national des arts et metiers - CNAM, 2019. Consulté le: 18 mai 2025. [En ligne]. Disponible sur: <https://theses.hal.science/tel-02180882>
- [11] L. Lac, « Méthodes de vision par ordinateur et d'apprentissage profond pour la localisation, le suivi et l'analyse de structure de plantes : application au désherbage de précision », phdthesis, Université de Bordeaux, 2022. Consulté le: 18 mai 2025. [En ligne]. Disponible sur: <https://theses.hal.science/tel-03623086>
- [12] A. L. Floch, « Méthodes de localisation en intérieur en 5G », phdthesis, Université de Toulouse, 2025. Consulté le: 18 mai 2025. [En ligne]. Disponible sur: <https://theses.hal.science/tel-04985094>
- [13] M. Hamid et R. Rouam, « Optimisation de déploiement et de localisation de cible dans les réseaux de capteurs sans fil », Thesis, Université Akli Mohand Oulhadje-

- Bouira, 2019. Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <http://dspace.univ-bouira.dz:8080/jspui/handle/123456789/10312>
- [14] A. Mesmoudi, « La localisation dans les réseaux de capteurs sans fil », Thesis, 2016. Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <http://dspace1.univ-tlemcen.dz/handle/112/18052>
- [15] T. Perković, L. Dujčić Rodić, J. Šabić, et P. Šolić, « Machine Learning Approach towards LoRaWAN Indoor Localization », *Electronics*, vol. 12, n° 2, Art. n° 2, janv. 2023, doi: 10.3390/electronics12020457.
- [16] M. Al mojamed, « On the Use of LoRaWAN for Mobile Internet of Things: The Impact of Mobility », *Appl. Syst. Innov.*, vol. 5, n° 1, Art. n° 1, févr. 2022, doi: 10.3390/asi5010005.
- [17] B. C. Fargas et M. N. Petersen, « GPS-free geolocation using LoRa in low-power WANs », in *2017 Global Internet of Things Summit (GIoTS)*, juin 2017, p. 1-6. doi: 10.1109/GIOTS.2017.8016251.
- [18] N. Podevijn *et al.*, « TDoA-Based Outdoor Positioning with Tracking Algorithm in a Public LoRa Network », *Wirel. Commun. Mob. Comput.*, vol. 2018, n° 1, p. 1864209, 2018, doi: 10.1155/2018/1864209.
- [19] A. Dongare *et al.*, « OpenChirp: A Low-Power Wide-Area Networking architecture », in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, mars 2017, p. 569-574. doi: 10.1109/PERCOMW.2017.7917625.
- [20] F. Wolf, C. Villien, S. de Rivaz, F. Dehmas, et J.-P. Cances, « Improved multi-channel ranging precision bound for narrowband LPWAN in multipath scenarios », in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, avr. 2018, p. 1-6. doi: 10.1109/WCNC.2018.8377203.
- [21] W. Bakkali, M. Kieffer, M. Lalam, et T. Lestable, « Kalman filter-based localization for Internet of Things LoRaWAN™ end points », in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, oct. 2017, p. 1-6. doi: 10.1109/PIMRC.2017.8292242.
- [22] K.-H. Lam, C.-C. Cheung, et W.-C. Lee, « LoRa-based localization systems for noisy outdoor environment », in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, oct. 2017, p. 278-284. doi: 10.1109/WiMOB.2017.8115843.
- [23] K.-H. Lam, C.-C. Cheung, et W.-C. Lee, « New RSSI-Based LoRa Localization Algorithms for Very Noisy Outdoor Environment », in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, juill. 2018, p. 794-799. doi: 10.1109/COMPSAC.2018.10340.
- [24] M. Aernouts, R. Berkvens, K. Van Vlaenderen, et M. Weyn, « Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas », *Data*, vol. 3, n° 2, Art. n° 2, juin 2018, doi: 10.3390/data3020013.
- [25] Z. He, Y. Li, L. Pei, et K. O'Keefe, « Enhanced Gaussian Process-Based Localization Using a Low Power Wide Area Network », *IEEE Commun. Lett.*, vol. 23, n° 1, p. 164-167, janv. 2019, doi: 10.1109/LCOMM.2018.2878704.
- [26] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, H. K. Qureshi, et M. Gidlund, « RSSI Fingerprinting-Based Localization Using Machine Learning in

- LoRa Networks », *IEEE Internet Things Mag.*, vol. 3, n° 4, p. 53-59, déc. 2020, doi: 10.1109/IOTM.0001.2000019.
- [27] H. Chen *et al.*, « Research on indoor positioning method based on LoRa-improved fingerprint localization algorithm », *Sci. Rep.*, vol. 13, n° 1, p. 13981, août 2023, doi: 10.1038/s41598-023-41250-x.
- [28] K. Lu, Y. Yue, et J. Ma, « Enhanced LoRaWAN RSSI Indoor Localization Based on BP Neural Network », in *2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, sept. 2021, p. 190-195. doi: 10.1109/ICISCAE52414.2021.9590790.
- [29] W. Ingabire, H. Larijani, R. M. Gibson, et A.-U.-H. Qureshi, « LoRaWAN Based Indoor Localization Using Random Neural Networks », *Information*, vol. 13, n° 6, Art. n° 6, juin 2022, doi: 10.3390/info13060303.
- [30] J. Purohit, X. Wang, S. Mao, X. Sun, et C. Yang, « Fingerprinting-based Indoor and Outdoor Localization with LoRa and Deep Learning », in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, déc. 2020, p. 1-6. doi: 10.1109/GLOBECOM42002.2020.9322261.
- [31] D. J. Suroso, A. S. H. Rudianto, M. Arifin, et S. Hawibowo, « Random Forest and Interpolation Techniques for Fingerprint-based Indoor Positioning System in Un-ideal Environment », *Int. J. Comput. Digit. Syst.*, vol. 10, n° 1, p. 701-713, mai 2021, doi: 10.12785/ijcds/100166.
- [32] I. T. Ali, A. Muis, et R. F. Sari, « A DEEP LEARNING MODEL IMPLEMENTATION BASED ON RSSI FINGERPRINTING FOR LORA-BASED INDOOR LOCALIZATION », *EUREKA Phys. Eng.*, n° 1, Art. n° 1, janv. 2021, doi: 10.21303/2461-4262.2021.001620.
- [33] H. Zhu *et al.*, « Extreme RSS Based Indoor Localization for LoRaWAN With Boundary Autocorrelation », *IEEE Trans. Ind. Inform.*, vol. 17, n° 7, p. 4458-4468, juill. 2021, doi: 10.1109/TII.2020.2996636.
- [34] A. Augustin, J. Yi, T. Clausen, et W. M. Townsley, « A Study of LoRa: Long Range & Low Power Networks for the Internet of Things », *Sensors*, vol. 16, n° 9, Art. n° 9, sept. 2016, doi: 10.3390/s16091466.
- [35] U. Raza, P. Kulkarni, et M. Sooriyabandara, « Low Power Wide Area Networks: An Overview », *IEEE Commun. Surv. Tutor.*, vol. 19, n° 2, p. 855-873, 2017, doi: 10.1109/COMST.2017.2652320.
- [36] « LoraWan | anteny i akcesoria ». Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://interline.pl/pl/Information-and-Tips/LoraWan>
- [37] M. Hammache, « Performances et Gestion de l'itinérance dans les réseaux LoRaWAN », phdthesis, Institut National Polytechnique de Toulouse - INPT, 2022. Consulté le: 19 mai 2025. [En ligne]. Disponible sur: <https://theses.hal.science/tel-04248270>
- [38] N. Benkahla, « Gestion de la qualité de service (QoS) dans un réseau LoRaWAN avec mobilité », phdthesis, Ecole Supérieure des Communications de Tunis, 2021. Consulté le: 19 mai 2025. [En ligne]. Disponible sur: <https://hal.science/tel-03283203>
- [39] V. Gache, « Passerelle LoRa : La révolution de la connectivité longue distance », *informatique Industrielle & IIoT*. Consulté le: 19 mai 2025. [En ligne]. Disponible

sur: <https://blog.integral-system.fr/passerelle-lora-la-revolution-de-la-connectivite-longue-distance/>

- [40] « LoRa: Symbol Generation ». Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <http://www.sghoslya.com/p/lora-is-chirp-spread-spectrum.html>
- [41] M. Kamal, M. Alam, A. Abu, et M. Su'ud, « SNR and RSSI Based an Optimized Machine Learning Based Indoor Localization Approach: Multistory Round Building Scenario over LoRa Network », *Comput. Mater. Contin.*, vol. 80, n° 2, p. 1927-1945, 2024, doi: 10.32604/cmc.2024.052169.
- [42] R. A. R. Antayhua, M. D. Pereira, N. C. Fernandes, et F. Rangel de Sousa, « Exploiting the RSSI Long-Term Data of a WSN for the RF Channel Modeling in EPS Environments », *Sensors*, vol. 20, n° 11, Art. n° 11, janv. 2020, doi: 10.3390/s20113076.
- [43] « Glossaire - Intelligence artificielle - www.coe.int », Intelligence artificielle. Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://www.coe.int/fr/web/artificial-intelligence/glossary>
- [44] « K Plus Proches Voisins / K-Nearest Neighbors /KNN | PDF | Régression linéaire | Mathématiques appliqués ». Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://fr.scribd.com/document/619772332/f981be00-176f-42e7-9d53-d65a9d656014>
- [45] S. J. Saleh, S. Q. Ali, et A. M. Zeki, « Random Forest vs. SVM vs. KNN in classifying Smartphone and Smartwatch sensor data using CRISP-DM », in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, oct. 2020, p. 1-4. doi: 10.1109/ICDABI51230.2020.9325607.
- [46] « Anas Brital | Random Forest Algorithm Explained . » Consulté le: 30 mai 2025. [En ligne]. Disponible sur: <https://anasbrital98.github.io/blog/2021/Random-Forest/>
- [47] A. Lorberfeld, « Machine Learning Algorithms In Layman's Terms, Part 1 », TDS Archive. Consulté le: 10 mai 2025. [En ligne]. Disponible sur: <https://medium.com/data-science/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b>
- [48] S. Santos, « ESP32 with LoRa using Arduino IDE | Random Nerd Tutorials ». Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://randomnerdtutorials.com/esp32-lora-rfm95-transceiver-arduino-ide/>
- [49] « Rapport Esp32 | PDF | Microcontrôleur | Entrée/Sortie ». Consulté le: 6 juin 2025. [En ligne]. Disponible sur: <https://fr.scribd.com/document/786738515/RAPPORT-ESP32-1>
- [50] « esp32\_technical\_reference\_manual\_en.pdf ». Consulté le: 2 juin 2025. [En ligne]. Disponible sur: [https://www.espressif.com/sites/default/files/documentation/esp32\\_technical\\_reference\\_manual\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf)
- [51] « lora-ra-02.jpg (504×368) ». Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://tizitronic.com/wp-content/uploads/2024/04/lora-ra-02.jpg>
- [52] « SX1276 ». Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://www.semtech.com/products/wireless-rf/lora-connect/sx1276>
- [53] « Logiciel : IDE – Arduino : l'essentiel ». Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://arduino.blaisepascal.fr/presentation/logiciel/>

- [54] « Comment utiliser la communication SPI avec un Arduino ? », Tutoriels Arduino & Raspberry Pi. Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://www.tutoriel-arduino.com/communication-spi-arduino/>
- [55] « GitHub - sandeepmistry/arduino-LoRa: An Arduino library for sending and receiving data using LoRa radios. », GitHub. Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://github.com/sandeepmistry/arduino-LoRa>
- [56] « 04-PI-4-Detailed-2.png (1200×844) ». Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://www.framboise314.fr/wp-content/uploads/2019/06/04-PI-4-Detailed-2.png>
- [57] « Qu'est-ce que le Raspberry Pi ? » Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://www.kubii.com/fr/content/72-qu-est-ce-que-le-raspberry-pi?srsltid=AfmBOoq5i48HzQPRu1JoNXk02hIcAlMN9dl92eRYFuvQNJi0f-YC6me0>
- [58] F. MOCQ, « Raspberry Pi 4 : 4 nouveautés qui vont vous faire craquer ! », Framboise 314, le Raspberry Pi à la sauce française.... Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://www.framboise314.fr/raspberry-pi-4-4-nouveautes-qui-vont-vous-faire-craquer/>
- [59] L. Alphorm, « Pourquoi Choisir Thonny pour Python Débutant », Blog Alphorm. Consulté le: 2 juin 2025. [En ligne]. Disponible sur: <https://blog.alphorm.com/thonny-apprentissage-python-debutant>
- [60] « Qu'est-ce que le langage de programmation Python ? | Teradata ». Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://www.teradata.com/insights/data-platform/what-is-python-programming-language>
- [61] *Inteform/PyLora*. (11 avril 2025). C. Inteform. Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://github.com/Inteform/PyLora>
- [62] « 29.1. sys — Paramètres et fonctions propres à des systèmes — Documentation Python 3.6.15 ». Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://docs.python.org/fr/3.6/library/sys.html>
- [63] « struct — Interpret bytes as packed binary data », Python documentation. Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://docs.python.org/3/library/struct.html>
- [64] « draft\_spidev\_doc\_fr.pdf ». Consulté le: 3 juin 2025. [En ligne]. Disponible sur: [https://www.sigmdel.ca/michel/ha/rpi/dnld/draft\\_spidev\\_doc\\_fr.pdf#page=1.25](https://www.sigmdel.ca/michel/ha/rpi/dnld/draft_spidev_doc_fr.pdf#page=1.25)
- [65] « Connecteur MariaDB/Python — Documentation du connecteur MariaDB/Python 1.1.12 ». Consulté le: 3 juin 2025. [En ligne]. Disponible sur: <https://mariadb-corporation.github.io/mariadb-connector-python/index.html>
- [66] O. Shalev (@ofirdi), « Recall, Precision, F1, ROC, AUC, and everything », The Startup. Consulté le: 8 juin 2025. [En ligne]. Disponible sur: <https://medium.com/swlh/recall-precision-f1-roc-auc-and-everything-542aedef322b9>

