



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامعة عباس لغرور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الصناعية



N° Série :

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Télécommunications

Spécialité : Systèmes des Télécommunications

THEME

**Classification de l'environnement intérieur
à base des mesures du signal WiFi**

Réalisé par : - TAKOUACHET Dounya

- BELHABES Rahma

Soutenu le : 15/07/2021 *Devant le jury composé de:*

Mme. MEDJALDI Malika

Mr. DOUAK Fouzi

Mr. BEDRA Sami

Présidente

Encadreur

Examineur

Université Abbes Laghrour-Khenchela

Université Abbes Laghrour-Khenchela

Université Abbes Laghrour-Khenchela

Promotion 2020/2021

Je dédie ce travail à :

Ma mère,

Mon père,

Mes frères et mes sœurs,

Tous mes collègues de promotion,

Dounya

Je dédie ce mémoire à :
Mes parents, Mes frères et mes sœurs,
Tous mes collègues de promotion,

Rahma

Remerciements

Nous tenons à remercier ALLAH tout puissant qui nous a donné la santé, le courage, et la volonté pour réaliser ce travail.

*Nous remercions notre encadreur Monsieur Dr. **Fouzi Douak**, enseignant à l'université de Khenchela, pour son suivi, pour son aide précieux.*

*Nous adressons nos plus sincères remerciements à Madame Dr. **Malika Medjaldi**, enseignante à l'université de Khenchela, pour nous avoir fait l'honneur de présider le jury de notre mémoire. Nous remercions aussi au Monsieur Dr. **Sami Bedra**, enseignant à l'université de Khenchela, qui a accepté d'examiner ce travail.*

Nous tenons à remercier particulièrement tous les enseignants et les responsables de notre département, qui ont veillé sur notre formation.

Nos remerciements vont également à tout le personnel du département du génie industriel pour leur gentillesse et leur disponibilité.

Enfin, nous exprimons notre remerciement à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Résumé

Les applications de localisation intérieure ont des besoins spécifiques en termes de précision, auxquels une technologie de localisation GPS ne peut généralement pas répondre. En combinant les mesures de la qualité du signal WiFi avec les techniques d'intelligence artificielle, on fournit une nouvelle solution robuste pour la localisation d'un utilisateur dans un environnement fermé, ne nécessitant pas d'infrastructures dédiées. L'objectif de ce travail consiste à la simulation et la réalisation d'un système de localisation en intérieur avec la technologie WiFi en utilisant la qualité du signal RSSI et les techniques de l'intelligence artificielle. Notre proposition est soutenue par des tests qui ont donné des résultats acceptables.

Mots clés: Localisation en intérieur, GPS, WiFi, RSSI, Intelligence artificielle.

Abstract

Indoor location-based applications have strong precision requirements that usually cannot be met by the GPS location-sensing technology. Combining the strong identification of WiFi signal quality measurements with artificial intelligence techniques provides a powerful new solution for indoor location sensing, without requiring dedicated infrastructures. The objective of this work is the simulation and realization of an indoor location system with WiFi technology using the quality of signal RSSI combined with artificial intelligence techniques. Our proposal is supported by tests which have given acceptable results.

Keywords: Indoor location, WiFi, RSSI, Artificial intelligence.

ملخص

تحتوي التطبيقات القائمة على تحديد المواقع الجغرافية داخل هياكل المؤسسات (مواقع داخلية) على متطلبات دقة عالية لا يمكن تحقيقها عادةً بواسطة نظام تحديد المواقع العالمي (GPS). يوفر الجمع بين قياسات جودة الإشارة WiFi وتقنيات الذكاء الاصطناعي حلاً جديداً لتحديد المواقع الداخلية ، دون الحاجة إلى بنى تحتية مخصصة لذلك. الهدف من هذا العمل هو محاكاة وتحقيق نظام لتحديد الموقع الجغرافي الداخلي باستخدام تقنية WiFi و ذلك باستخدام جودة الإشارة RSSI جنباً إلى جنب مع تقنيات الذكاء الاصطناعي . اقترحنا مدعوم بعدة اختبارات والتي أعطت نتائج مقبولة.

الكلمات المفتاحية: الموقع الجغرافي الداخلي , WiFi, RSSI , الذكاء الاصطناعي.

Liste des tableaux

Tableau II. 1. Fonctions de transfert $a=f(n)$	25
Tableau III. 1. Exemple de la base de données pour la localisation.....	46
Tableau III. 2. Architecture du modèle MLP.	51
Tableau III. 3. Variation des paramètres à optimisés par le modèle SVM.....	55
Tableau IV. 1. Affectation des broches GPS.	63
Tableau IV. 2. Commandes et les rôles d'un programme GPS.	65
Tableau IV. 3. Connexions des broches LCD de communication.	71

Liste des figures

Figure I. 1. Position en fonction du nombre et du type de nœuds.....	6
Figure I. 2. Processus de localisation.	7
Figure I. 3. Technologies de localisation indoor.	10
Figure I. 4. Système avec plusieurs transmetteurs et un dispositif de localisation.	11
Figure I. 5. Système avec un dispositif de localisation associé à chacune des palettes.	12
Figure I. 6. Segment spatial.	13
Figure I. 7. Emplacement de segments de Contrôles dans le monde.	14
Figure I. 8. Segment utilisateur.	15
Figure I. 9. Mode infrastructure.	18
Figure I. 10. Mode Ad-Hoc.	19
Figure II. 1. Neurone artificiel.	21
Figure II. 2. Mise en correspondance neurone biologique / neurone artificiel.	22
Figure II. 3. Structure d'un réseau de neurone multicouche.	23
Figure II. 4. Fonction de transfert : (a) du neurone "seuil"; (b) du neurone "linéaire", (c) "sigmoïde", et (d) du neurone "tangente hyperbolique"	24
Figure II. 5. Couche de S neurones.	26
Figure II. 6. Représentation matricielle d'une couche de S neurones.....	27
Figure II. 7. Représentation matricielle d'un réseau de trois couches.	27
Figure II. 8. Architecture générique d'un RBFN.	31
Figure II. 9. Exemple d'un hyperplan séparateur.....	34
Figure II. 10. Exemple de vecteurs de support.....	35
Figure II. 11. Exemple de marge maximal (hyperplan valide).	35
Figure II. 12. Transformer l'espace de départ vers un espace de dimension supérieure.	36
Figure II. 13. a) Cas linéairement séparable, b) Cas non linéairement séparable.	37
Figure II. 14. Approche une-contre-une.....	40
Figure II. 15. Séparation entre trois classes en utilisant la méthode 1vsR.....	41
Figure II. 16. Graphe de décision acyclique orienté à quatre classes.....	42

Figure II. 17. SVM multiclasse par arbre de décision.....	42
Figure III. 1. Exemples de la base de données.....	47
Figure III. 2. Exemple de la division de la base de données en deux ensembles : Apprentissage et test.....	47
Figure III. 3. Localisation par FingerPrint dans un réseau WiFi.	48
Figure III. 4. Précision de classifieur MLP sur l'ensemble de validation (<i>Acc</i>) en fonction des nombres de neurones dans les couches cachées.	51
Figure III. 5. Précision de classifieur MLP sur l'ensemble de validation (<i>Acc</i>) en fonction des nombres de neurones dans la première couche cachée et le paramètre d'accélération de réseaux de neurone momentum.	52
Figure III. 6. Précision de classifieur MLP sur l'ensemble de validation (<i>Acc</i>) en fonction des nombres de neurones dans la première couche cachée et le taux d'apprentissage.....	52
Figure III. 7. Matrice de confusion sur l'ensemble de test (MLP).....	53
Figure III. 8. Précision (<i>Acc</i>) de réseaux de neurones RBFN en fonction de nombre de neurones dans la couche cachée et la largeur de gaussienne σ	54
Figure III. 9. Matrice de confusion de la meilleure architecture RBFN sur l'ensemble de test.	54
Figure III. 10. Précision (<i>Acc</i>) en fonction de C et γ	55
Figure III. 11. Matrice de confusion sur l'ensemble de test (SVM).....	56
Figure III. 12. Etude comparative entre MLP, RBFN et SVM.....	56
Figure IV. 1. Carte Arduino Uno.	60
Figure IV. 2. Description de l'Interface Arduino IDE.	61
Figure IV. 3. Structure d'un programme Arduino.....	62
Figure IV. 4. Module GPS (GY-NEO6MV3).	63
Figure IV. 5. Branchement de module GPS avec Arduino Uno.	63
Figure IV. 6. Circuit électrique du module GPS avec la carte Arduino Uno.	64
Figure IV. 7. (a) Montage de la localisation GPS, (b) affichage de moniteur série de l'arduino sur PC.	66
Figure IV. 8. Affichage de moniteur série de l'arduino sur PC, dans le cas d'un environnement extérieur.	67

Figure IV. 9. La localisation outdoor d'un utilisateur à l'aide de Google Maps.....	67
Figure IV. 10. La carte ESP32 DEVKIT WROOM.....	68
Figure IV. 11. Les Caractéristique de La carte ESP32 DEVKIT WROOM.	68
Figure IV. 12. Schéma récapitulatif du système de positionnement basé sur ESP32.	70
Figure IV. 13. Assemblage du ESP32WROOM et LCD 16x2	71
Figure IV. 14. Circuit électrique du module ESP32WROOM et LCD 16x2.....	72
Figure IV. 15. Affichage de la qualité du signal (RSSI).	72
Figure IV. 16. RSSI des réseaux disponibles à proximité d'ESP32.....	73
Figure IV. 17. Logo du système d'exploitation Raspbian et Windows 10 IoT.....	74
Figure IV. 18. Adresse IP de la carte Raspberry <i>Pi</i>	74
Figure IV. 19. Fichier de configuration réseau.....	75
Figure IV. 20. Logiciel Putty.....	76
Figure IV. 21. Invite de commande via SSH.....	77
Figure IV. 22. Mesure de la qualité du signal (RSSI) à base de Raspberry <i>Pi</i>	78
Figure IV. 23. Connectez Raspberry <i>Pi</i> au <i>PC</i>	79
Figure IV. 24. Capture d'écran des RSSI sur Raspberry <i>Pi</i> via un terminal.....	79

Liste des symboles

AoA: L'angle d'arrivée

BSS: Basic service set

BSSID: Basic service set Identifier

DS: Distribution system

DHCP: Dynamics Host Configuration Protocol

DoD: Département de défense américain

EGNOS: European Geostationary Navigation Overlay Service

ESS: Extended service Set

ESSID: Extended Service Set Identifier

EEPROM: Electrically Erasable Programmable Read-Only Memory

E/S: Entrée/Sortie

GPS: Global Positioning System.

GLONASS: Global Navigation Satellite System

GSM: Groupe Spécial Mobile

GPIO: General Purpose Input Output

IDE: Integrated Development Environment

IEEE: Electrical and Electronics Engineers

IBSS: Independent Basic Service Set

IP: Internet Protocol.

LUT: Local User Terminal

LED: Light Emitting Diode

LCD: Liquid Crystal Library

MAC: Media Access Control

MEOSAR: Medium Earth Orbit Search and Rescue

MCC: Mission Control Centers

MLP: Multi-Layer Perceptron

NN: Le nombre de nuerons.

PDA: Personale Digital Assistant

PA: Point D'Accès

RFID: Radio Frequency Identification
RSSI: Received Signal Strength Indicator
RCC: Rescue Coordination Centers
RBFN: Radial Basis Function Network
SAR: Search and rescue Service
SSH: Serveur Secure Shell.
SRAM: Static random access memory
SNR: Signal-to-Noise Ratio
ToA: le temps d'arrivée
TDOA: Time Difference Of Arrival
UWB: L'Ultra Wide band
WAAS: Wide Area Augmentation System
WiFi: Wireless Fidelity

Table des matières

Introduction Générale	1
Chapitre I	
Généralités sur les systèmes de localisation	
I.1. Introduction	5
I.2. Localisation	5
I.2.1. Définition de localisation indoor.....	6
I.2.2. Processus de localisation indoor	6
I.2.3. Les usage de localisation indoor	7
I.3. Technologies de localisation indoor.....	8
I.3.1. Technologie radio-identification RFID.....	8
I.3.2. Technologie Ultrason	9
I.3.3. Technologie Bluetooth	9
I.4. Développement du système de localisation	10
I.4.1. Premier modèle (un transmetteur sans-fil sur chacune des marchandises ou des palettes)	11
I.4.2. Deuxième modèle (un dispositif de localisation par client).....	12
I.5. Systèmes de localisation	12
I.5.1. Système GPS	12
I.5.1.1. Principe	12
I.5.1.2. Différents segments du système GPS.....	13
I.5.2. Système GALILEO	15
I.5.3. Système EGNOS	15
I.5.4. Système MEOSAR.....	16
I.5.5. Système GLONASS.....	16
I.5.6. Wifi	17
I.5.6.1. Les usages d'un réseau WiFi.....	17
I.5.6.2. Architecture réseau WIFI.....	17
I.6. Conclusion.....	19

Chapitre II

Classification par réseaux de neurones artificiels et machines à vecteurs de support

II.1. Introduction.....	21
II.2. Réseaux de neurones artificiels	21
II.2.1. Réseaux de neurones multicouche MLP.....	22
II.2.1.1. Fonctions de transfert.....	23
II.2.1.2. Architecture de réseau.....	26
II.2.1.3. Mise en œuvre des réseaux neuronaux	28
II.2.1.4. Algorithmes apprentissage.....	29
II.2.2. Réseaux à fonction radiale de base RBFN.....	30
II.2.2.1. Architecture de base.....	31
II.2.2.2. Caractéristiques de base.....	31
II.2.2.3. Variantes du RBFN.....	32
II.3. Machine à vecteurs de support.....	33
II.3.1. Principe général	34
II.3.2. Notions de base: Hyperplan, marge et support vecteur	34
II.4. Technique de validation.....	38
II.4.1. Validation simple	38
II.4.2. Monte-Carlo validation Croisé	38
II.4.3. <i>k-fold</i> cross-validation.....	38
II.5. Classification multiclasse.....	39
II.5.1. Une-contre-une (1vs1)	39
II.5.2. Une-contre-reste (1vsR).....	41
II.5.3. Graphe de décision.....	41
II.5.4. SVM basées arbres de décision	42
II.6. Conclusion	43

Chapitre III

Résultats de simulations

III.1. Introduction	45
III.2. Description de la base de données	45
III.3. Méthode proposée.....	48
III.4. Localisation par réseaux de neurones artificiels et machine à vecteurs de support	49
III.4.1. Évaluation des performances.....	50
III.4.1.1. Taux de classification	50
III.4.1.2. Matrice de confusion	50
III.4.2. Classification par réseaux de neurones artificiels.....	51
III.4.3. Classification par machine à vecteurs de support.....	55
III.4.4. Etude comparative	56
III.5. Conclusion	57

Chapitre IV

Réalisation pratique

IV. 1. Introduction	59
IV. 2. Système de localisation GPS.....	59
IV.2.1. Carte Arduino	59
IV.2.1.1. Partie Hardware (Le matériel).....	59
IV.2.1.2. Partie logiciel (Software).....	60
IV.2.2. Modules GPS.....	62
IV.2.2.1. Branchement du module GPS	63
IV.2.2.2. Programmation de Module GPS	64
IV.2.2.3. Exemple de programme	65
IV.2.2.4. Fonctionnement de montage GPS	66
IV. 3. ESP32	68
IV.3.1. Fonctionnement de montage ESP32.....	69
IV. 4. Raspberry <i>Pi</i>	73
IV.4.1. Aspects Logiciel	73
IV.4.2. Accès à distance au Raspberry <i>Pi</i>	74

IV.4.3. Choix de l'adresse IP du Raspberry <i>Pi</i>	74
IV.4.4. Connexion à distance via SSH	76
IV.4.5. Caractéristique de la carte Raspberry <i>Pi</i>	77
IV.4.6. Fonctionnement de montage à base de Raspberry <i>Pi</i>	78
IV. 5. Conclusion.....	80
Conclusion générale.....	82
<i>Bibliographie</i>	85

Introduction générale

Introduction générale

La localisation joue un rôle essentiel dans notre vie quotidienne, il est devenu indispensable pour se positionner aussi bien à l'intérieur qu'à l'extérieur des environnements (aéroport, bâtiments, campus et centres commerciaux, etc.). Les systèmes de positionnement sont utilisés dans plusieurs domaines, par exemple pour trouver l'hôtel ou le restaurant le plus proche, avoir des informations touristiques sur le lieu que l'on visite...etc. Dans le domaine médical par exemple, on utilise les systèmes de positionnement pour des personnes qui souffrent de pertes de mémoire ou plus généralement sur les personnes qui présentent un risque de ne pas retrouver leur chemin au cas où elles se perdraient [1].

Plusieurs systèmes de localisation ont été proposés dans la littérature scientifique: la localisation en utilisant des techniques intelligente à base de traitement de l'image [2, 3], le Li-Fi qui se base essentiellement sur la lumière pour la localisation, l'ultrason qui utilise la géolocalisation sonore inaudible pour l'homme, l'étiquette RFID composée de puces pour identifier les objets,... etc. [4, 5]. En plus, beaucoup de technologies complémentaires sont aujourd'hui disponibles pour répondre à des besoins différents. Le plus populaire est le système de navigation par satellite GPS (Global Positioning System), son utilisation dans les environnements intérieurs est limitée dû à la mauvaise pénétration des signaux GPS à l'intérieur des environnements «indoor». Ceci est dû aux signaux satellites faibles qui sont difficiles à recevoir dans les zones fermées, à cause des obstacles entre le satellite et le récepteur [2].

La plupart des solutions du problème de la localisation «indoor» se basent sur un mixage entre des signaux WiFi (Wireless Fidelity) et Bluetooth. La localisation «indoor» par le WiFi s'appuie essentiellement sur le réseau WiFi existant, il offre la possibilité de la localisation exacte d'un utilisateur grâce à la méthode d'empreinte radio (en anglais Radio Frequency Fingerprinting) [6].

Le WiFi est la technologie la plus utilisée pour la localisation en intérieur, en raison de sa disponibilité dans la plupart des environnements et de la propagation de son signal sur de longues distances. Les signaux WiFi sont couramment utilisés pour localiser un utilisateur mobile en faisant correspondre les positions spatiales et la force du signal enregistré à ces positions. Cependant, ces signaux peuvent fluctuer en raison de la propagation par trajets multiples et des interférences provenant d'autres appareils, ce qui affecte la précision de la localisation. Ce mémoire traite de la localisation indoor en utilisant les technologies WiFi et les techniques d'intelligence artificielle.

Dans le cadre de notre travail, l'indicateur de la puissance signal reçu RSSI (Received Signal Strength Indicator) des points d'accès WiFi est utilisée pour localiser la position de l'utilisateur mobile. Notre travail est divisée en deux phases, la première est le calibrage pour trouver les meilleurs paramètres de notre modèle hors ligne, et une deuxième phase qui été en temps réel permet de localisé un utilisateur dans un environnement fermé, en basant sur l'exploitation des paramètres extraits dans la première phase. Pour les deux phases, la localisation automatique est passée essentielles par trois étapes (i) la création des bases de données à partir des points d'accès WiFi (afficher et sauvegarder les valeurs de l'indicateur de puissance RSSI), cette phase consiste à relever les caractéristiques (Fingerprint) des signaux aux différents endroits d'un environnement intérieur (exemple : bâtiment), (ii) exploitées les caractéristique trouvé dans l'étape précédente pour la classification en utilisant les techniques d'intelligence artificielle: Réseaux de neurones multicouches (MLP) [7], Réseaux à fonction radiale de base (RBFN) [8], et Machine à vecteurs de support (SVM) [9, 10], (iii) la dernière étape est la localisation en temps réel, cette étape consiste à déterminer la position exacte de l'utilisateur en utilisant la meilleur architecture de l'intelligence artificielle trouvé dans l'étape de calibrage, plus le nombre de routeur WiFi augmente, plus la précision s'améliore; plusieurs mesures successives de puissance doivent être réalisées pour une meilleure évaluation de la position, les ondes radios étant affectées par plusieurs facteurs.

Le principal inconvénient des systèmes GPS est que ne fonctionne pas dans les zones intérieures. Pour résoudre ce problème, nous avons proposé d'utiliser deux cartes électroniques équipées de la technologie WiFi tels que ESP32 et le Raspberry Pi, pour l'objectif de la création des bases de données et la localisation indoor.

L'organisation générale du mémoire est décrite ci-dessous:

Dans **Le premier chapitre**, on va expliquer quelques notions de bases liées à notre projet; définition de localisation indoor, les usages de localisation indoor; ainsi que les différentes technologies de localisation indoor. Également on va citer les différents systèmes de localisation (GPS, WiFi,...).

Dans **Le deuxième chapitre**, on va présenter les différentes techniques de l'intelligence artificielle utilisé dans notre travail, tels que : les réseaux de neurones multicouche MLP, réseaux à fonction radiale de base RBFN et la méthode de machine à vecteurs de support (SVM).

Le troisième chapitre on va présenter les résultats de simulations de notre méthode de localisation.

Dans **Le quatrième chapitre** on va présenter et expliquer la conception et la réalisation matérielle et logicielle de notre travail, nous avons présenté les différentes étapes qui nous ont permis de réaliser un système de localisation indoor en utilisant les techniques de l'intelligence artificielle à base des mesures de la qualité du signal WiFi.

Enfin, la dernière partie de ce mémoire est consacrée à la **conclusion générale** et quelques **perspectives**.

Chapitre I

Généralités sur les systèmes de localisation

I.1. Introduction

Dans notre vie quotidienne, nous avons souvent besoin d'un moyen de localiser notre emplacement dans une zone spécifique comme les bâtiments et les hôpitaux. De tout temps, l'homme a développé des moyens techniques pour faciliter ses déplacements et augmenter son autonomie. Jusqu'à maintenant, les signaux de navigation GPS qui sont utilisés pour le positionnement à l'extérieur ne peuvent pas être bien captés à l'intérieur des bâtiments. Dans un tel environnement, le système GPS fonctionne en mode dégradé, ou ne fonctionne plus. Ces limites proviennent de : (i) Atténuation produite par les obstacles, les signaux reçus ont un faible SNR [Signal-to-Noise Ratio] (30db de moins qu'à l'extérieur). En outre, à chaque obstacle traversé correspond une atténuation de la puissance du signal. Ce qui implique une faible précision de positionnement. (ii) le positionnement à l'intérieur est soumis aux problèmes de multi-trajets, de masquage, des interférences... La superposition de différents signaux qui ont suivi différents chemins à l'antenne de réception, produit une distorsion de signal original.

Nous allons dans ce chapitre exposer quelques concepts de la localisation, puis les introduire le cadre théorique dans lequel s'inscrit notre travail, et par la suite nous présentons les éléments fondamentaux pour mesurer la qualité du signal WiFi sous une forme quantitative.

I.2. Localisation

Le terme désigne en premier lieu la position d'un objet sur la surface de la terre à l'aide d'un système de référence explicite, qui est souvent celui des coordonnées Géographiques qui sont indispensables pour localiser. Pour estimer la position d'un objet, il faut des points dont la position est connue ou des points qui connaissent leur propre position; points de référence ou ancres. Considérons qu'un nœud A désire estimer sa position. Son estimation de position dépendra du nombre et du type de position de nœuds de référence qu'il pourra trouver, comme les montrent la Figure I.1.

Dans le cas la figure I.1 (a), un seul nœud de référence est disponible. A aura une Position de type symbolique « je suis dans telle pièce »; Si A utilise deux points de références, Figure I.1 (b), alors son positionnement est Plus précis qu'avec un seul nœud, dans le cas où les nœuds de référence ont une très bonne Précision de localisation, alors A est l'un des points d'intersection des deux cercles ; La position de A est d'autant mieux estimée que le nombre d'ancres augmente. Dans le cas de la figure I.1 (c), la position de A est soit le point d'intersection des trois cercles, soit une aire de recouvrement limitée [11].

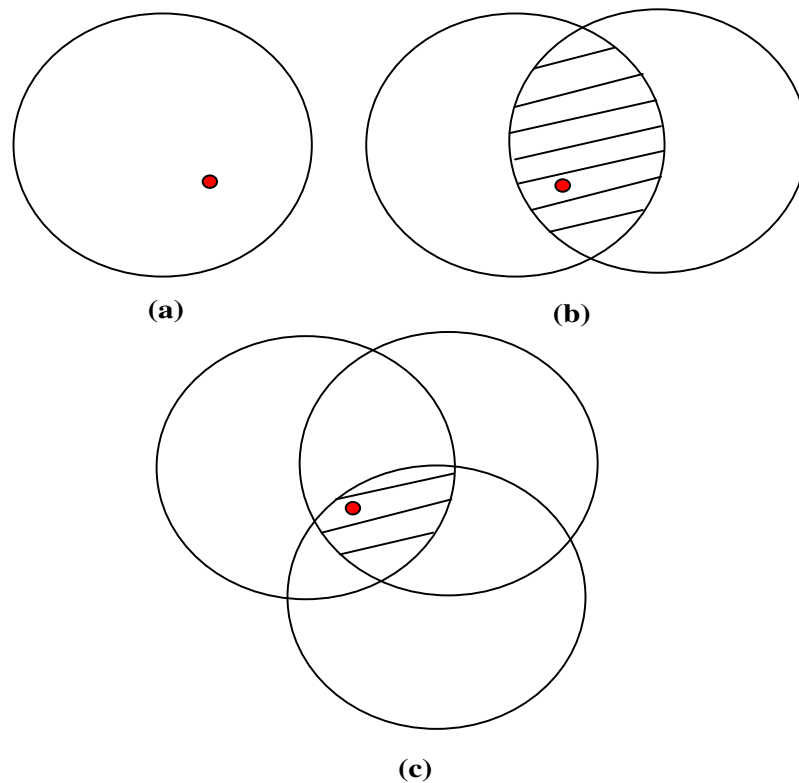


Figure I. 1. Position en fonction du nombre et du type de nœuds.

I.2.1. Définition de localisation indoor

Egalement connue sous les appellations «indoor location» ou «indoor positioning», la localisation indoor permet de connaître avec une précision plus ou moins grande, la localisation d'une personne ou d'un produit dans un espace ou un lieu fermé c'est-à-dire à l'intérieur des bâtiments. De nombreux usages sont imaginables, et certains sont même déjà mis en place. Parmi ceux-ci, on peut citer [12].

I.2.2. Processus de localisation indoor

La localisation indoor comme définie dans [13] est un processus qui permet de localiser en temps réel et de façon continue, un objet ou une personne dans un espace physique (un hôpital, une école, un musée, etc.).

Une autre définition plus succincte est donnée dans [13] et [14] indiquant que ce processus comporte généralement deux phases : caractérisation du signal et estimation de la position physique de la cible [15].

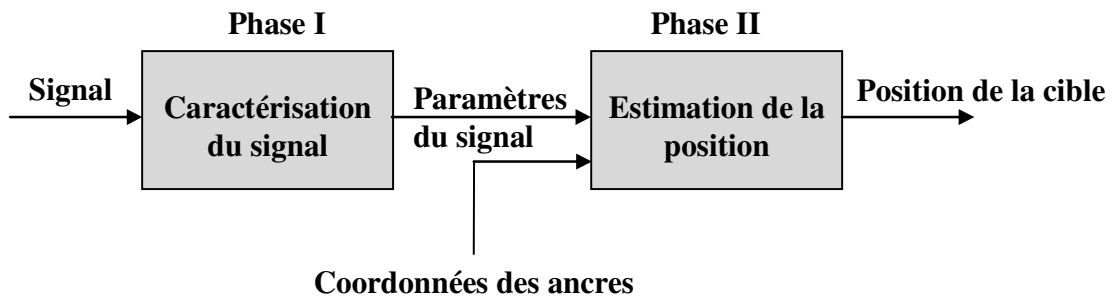


Figure I. 2. Processus de localisation.

Durant cette phase, le signal de communication échangé entre la cible et les ancrs de référence est étudié afin d'en extraire toute information utile pouvant servir comme paramètres d'entrée pour la seconde phase. Ces paramètres sont des métriques de caractérisation du signal communiqué. Ils sont soit le temps d'arrivée (ToA), soit la puissance du signal reçu (RSSI), soit l'angle d'arrivée (AoA), ces métriques seront détaillées dans les paragraphes suivants.

Phase II : Estimation de la position

Durant la seconde phase du processus de localisation, la position de la cible à localiser est déterminée en se basant sur les paramètres du signal extraits pendant la phase I. Plusieurs algorithmes sont employés : Triangulation, Trilatération, Fingerprinting,...

Les Soviétiques ont alors lancé un programme concurrent au GPS et qui se nomme GLONASS (Global Navigation Satellite System). Depuis, les Chinois essaient aussi de mettre en place leur propre système avec Beidou [Glo], et maintenant avec la mise en place de l'Union Européenne, Galileo sera disponible à partir de 2008-2010.

Ces différents systèmes de navigation par satellites sont abordés par la suite afin d'étudier les performances atteignables par chacun de ces systèmes en termes de localisation.

I.2.3. Les usage de localisation indoor

Aide à la navigation

Dans le cas d'une navigation allant de l'outdoor à l'indoor (ex : un client utilisant son application GPS pour se rendre de chez lui à un point d'intérêt précis situé à l'intérieur d'un centre commercial), celle-ci se doit d'être «sans coutures» : le mobile devra immédiatement détecter que l'utilisateur est rentré dans un bâtiment, et passer de manière transparente en mode navigation intérieure. Sites industriels Une fois un problème identifié sur un site industriel, une application mobile pourrait permettre de guider l'équipe technique mobile la plus proche vers la source du problème. Ensuite, une fonctionnalité d'aide à la résolution de

problème type Réalité augmentée ou Téléassistance pourrait prendre le relai si besoin.

Transports

Permettre à un voyageur de naviguer non seulement en extérieur, mais aussi à l'intérieur d'un lieu de type aéroport, gare ou station de métro.

Santé

Permettre au personnel de santé de retrouver des patients ou du matériel est un besoin souvent formulé. Concrètement, une application peut permettre la géo-localisation des nourrissons, des personnes souffrant d'Alzheimer, ou des appareils spécialisés égarés.

Marketing

Mais la principale application marketing serait le géomarketing : l'optimisation des revenus au mètre carré, en fonction des données de visite des clients (flux, temps de visite, zones de passages, zones de transformation, etc.). Il est possible de personnaliser des offres marketing, non seulement selon les données connues sur le client par la marque (carte de fidélité) mais aussi selon le contexte géographique d'un client en magasin [12].

I.3. Technologies de localisation indoor

Un grand nombre de concepts et technologies ont aujourd'hui démontré des possibilités prometteuses pour réaliser une localisation indoor en se basant sur le réseau de nœuds émetteurs. Les techniques possibles sont les réseaux de capteurs (Radio Frequency Identification : RFID passive, infrarouge, ultrason), les réseaux de communication (WiFi, Bluetooth, RFID active, ZigBee...) et les réseaux mobiles (DECT-DPS, GSM « Groupe Spécial Mobile », Cell ID...), sachant que chacune d'entre elles présente sa propre solution de géolocalisation indoor et essaye de satisfaire l'utilisateur. Il existe deux classes méthodes de localisation [12].

I.3.1. Technologie radio-identification RFID

La radio-identification connue sous l'acronyme RFID est une méthode d'identification automatique qui permet de mémoriser et de récupérer des données à distance en utilisant des marqueurs appelés «radio-étiquettes».

Un système RFID est composé essentiellement des deux éléments suivants [15]:

- ✚ **Tag RFID**, aussi appelé étiquette radio, marqueur radio, identifiant, transpondeur,... Il s'agit d'un récepteur incorporé sur des objets, des animaux ou mêmes des humains. Chaque «tag» est muni d'une puce qui sert à mémoriser les données. Cette puce est reliée à une antenne qui assure les échanges de ces données. Selon l'intégration ou non d'un émetteur radio, les tags RFID peuvent être classés en deux catégories :

- **Tag RFID passif:** ce type de tag n'intègre pas d'émetteur radio. Ne comportant pas une source d'énergie, le tag passif utilise souvent l'onde radio reçue de l'interrogateur afin d'alimenter son circuit électronique. Ce type tag est utilisé dans plusieurs applications de capture simple de données sans nécessité de les émettre (capture de température, etc.).
- **Tag RFID actif:** ce tag comporte un émetteur radio. Il embarque généralement une source d'alimentation en énergie (batterie, pile..). Cette source d'énergie assure la fonction diffusion du signal vers un lecteur RFID. Ce type de tag est largement utilisé dans des applications de localisation de personnes, de véhicules et la traçabilité d'objets de tout genre.

✚ **Interrogateurs RFID,** ont comme vocation l'activation des tags RFID en leur fournissant l'énergie dont ils ont besoin.

I.3.2. Technologie Ultrason

Se base essentiellement sur les ondes sonores envoyées, dont la fréquence est supérieure à 20000 Hz. L'écholocalisation consiste à envoyer des sons à diverse fréquences et à les récupérer.

La durée prise pour que l'onde revienne, et la nature des ondes renvoyées parmi toutes celles émises permettent de localiser les éléments. Cette technologie répond à une localisation indoor en temps réel en se basant sur la méthode TDOA et donne une précision de localisation inférieur à 1 mètre avec une consommation modérée quoi que le coût de mise en place de l'infrastructure est élevé et d'un autre coté cette technologie est très sensible à l'environnement (température, humidité...) et même l'écho ce qui peut endommager la précision de la localisation.

I.3.3. Technologie Bluetooth

Les iBeacon est un protocole créé par Apple es qui est devenu compatible avec Android et Microsoft. Il repose sur la technologie Bluetooth LE 4.0. À travers ce Bluetooth les Smartphones peuvent entrer en relation avec des petits transmetteurs sans fil sous la forme d'un petit boîtier appelé «Beacon» (balise). Ce dernier émet en permanence une petite quantité d'informations afin d'entrer en interaction avec les Smartphones présents dans une zone.



Figure I. 3. Technologies de localisation indoor.

I.4. Développement du système de localisation

Le développement du système de localisation a été séparé en 3 parties à étudier en parallèle : Premièrement A) communication entre le dispositif de localisation et l'extérieur du camion. Deuxièmement, B) application d'un algorithme de localisation pour le suivi des marchandises en temps réel. Troisièmement C) collecte de l'information de la présence des marchandises à l'aide de modules transmetteurs et récepteurs à l'intérieur du camion.

Dans l'industrie concernée, deux types de camions sont utilisés, fourgon sec (dry freight van) et camion réfrigéré (refrigerated freight van). La majorité du temps, l'industrie utilise le fourgon sec pour la livraison. Le fourgon sec permet une meilleure propagation pour les signaux vers l'extérieur que le camion réfrigéré. Cela est dû parce que le camion réfrigéré comporte le toit, le plancher et les murs en aluminium, tandis que le fourgon sec comporte un plancher métallique avec certaines parties en bois et certaines fois un toit en plastique. Puisque l'aluminium ne laisse pas passer les signaux et les réfléchit, le camion réfrigéré est plus isolé et fermé en termes de rayonnement électromagnétique. Les deux types de camions mesurent environ 16 m de longueur et se déplacent à une certaine vitesse. Une étude sera faite pour déterminer s'il est possible de communiquer avec l'extérieur du camion et s'il faudrait amplifier les signaux car le comportement de la remorque du camion sur les signaux peut être vu de manière semblable à celui d'une cage de Faraday. Pour l'intérieur de la remorque du camion, deux pistes sont étudiées; la première consiste à l'utilisation de la technologie d'identification par radiofréquence (RFID) et la deuxième consiste à l'utilisation de modules transmetteurs qui seraient installées sur chaque marchandise ou palette. De plus, deux

modèles de communication marchandise–‘équipement terminal’ sont étudiés dans cette recherche ; un premier modèle comprenant un transmetteur sans-fil sur chacune des palettes et un dispositif de localisation qui permet de gérer les informations qui proviennent d’eux et un deuxième modèle comprenant un dispositif de localisation sur chacune des palettes [16].

I.4.1. Premier modèle (un transmetteur sans-fil sur chacune des marchandises ou des palettes)

Dans le premier modèle, un transmetteur se trouve accroché sur chacune des marchandises tel que montré sur la Figure I.4. Durant le trajet, le dispositif de localisation reçoit l’information des transmetteurs sans-fil et la transmet à une station de base pour permettre aux propriétaires des avoir si leurs marchandises sont dans le camion et leurs coordonnées. Chaque marchandise contient un module qui permet de communiquer sa présence à un dispositif de localisation cellulaire durant le transport. L’avantage de ces modules est de savoir que lorsqu’une partie des marchandises est transféré dans un autre camion, le client sait si chacune d’entre elles est transférée. Les transmetteurs qui sont collées sur les palettes envoient l’information à un ou à quelques dispositifs de localisation selon le nombre de clients. Ces dispositifs transmettent ensuite l’information aux stations de base compatibles. Ces stations retransmettent cette information dans un réseau cellulaire pour informer le client. La fréquence pour la communication à l’intérieur (Fréquence module) est différente des fréquences de communication utilisées pour l’extérieur du camion (Fréquences cellulaires). Les ‘Fréquences cellulaires’ de la figure représentent plusieurs fréquences car les technologies cellulaires utilisent très souvent plus qu’une bande de fréquence. Un codage devrait donc être implémenté pour ces types de communication; à l’intérieur et avec l’extérieur du camion pour permettre la différenciation des palettes. Actuellement, il n’existe pas un système qui fait ce travail sur le marché au meilleur de la connaissance.

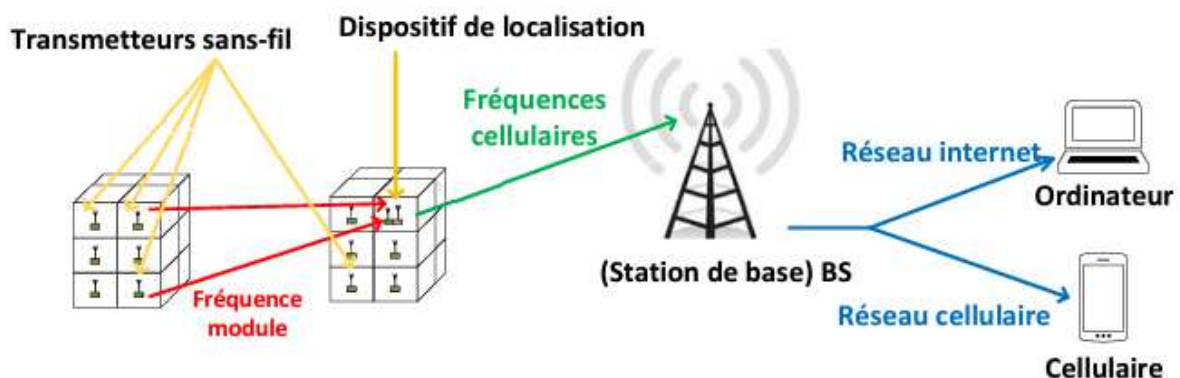


Figure I. 4. Système avec plusieurs transmetteurs et un dispositif de localisation.

I.4.2. Deuxième modèle (un dispositif de localisation par client)

Contrairement au premier modèle, dans celui-ci (Figure I.5), il y a un dispositif de localisation accroché sur chacune des palettes et il transmet l'information directement à la station sans passer par un autre module. Donc, il n'y a pas de communication entre les modules qui se trouvent à l'intérieur du camion. Ce modèle serait suffisant pour l'industrie puisque le dispositif de localisation est plus prioritaire qu'un système avec des transmetteurs. Il existe quelques modèles similaires sur le marché tel que Trakdot, LugLoc et PT300.

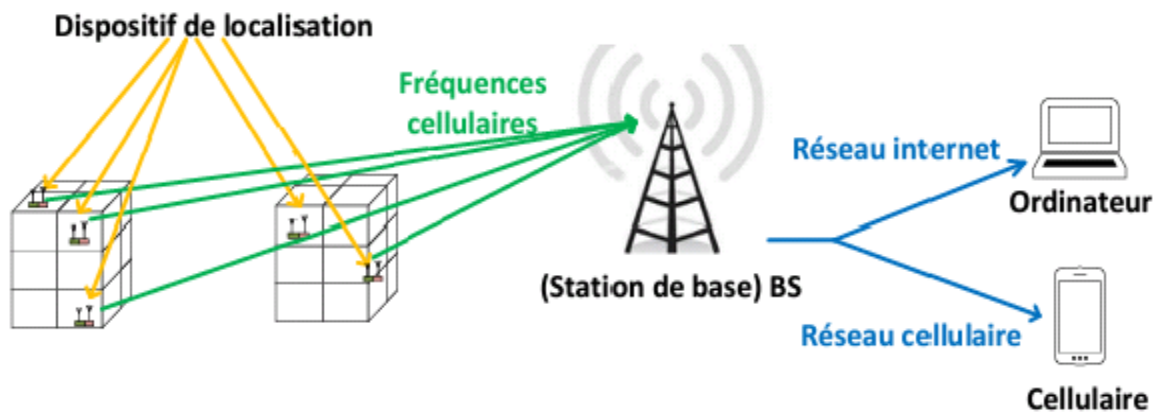


Figure I. 5. Système avec un dispositif de localisation associé à chacune des palettes.

I.5. Systèmes de localisation

I.5.1. Système GPS

Le système GPS est le plus populaire des systèmes de positionnement par satellites. Créé par le département de défense américain (DoD), le système GPS a pour mission de couvrir toute la terre grâce à une constellation comptant 24 satellites [17]. Les satellites GPS sont répartis sur six orbites, chaque orbite compte quatre satellites GPS. Ce système satellitaire fournit aux récepteurs GPS des informations en continue. Le récepteur mesure le temps de propagation de l'onde reçue et calcule ainsi la distance qui le sépare du satellite. Il répète cette opération pour chaque satellite en vue. Une fois que les distances entre le récepteur et les satellites sont connues, il devient possible de calculer la position par triangulation. Nous avons besoin de trois satellites pour déterminer un point et quatre satellites pour avoir une position en 3D (incluant la profondeur). Nous pouvons déduire que plus il y a de satellites GPS visibles au récepteur, plus la position calculée est exacte [18].

I.5.1.1. Principe

Le récepteur GPS va donc capter les signaux d'au moins quatre satellites et calculer sa position en fonction des informations reçues.

L'utilisation de cette technique nécessite l'installation d'un module GPS, dans le mobile ce qui est plus coûteux que les autres moyens utilisés pour la géo localisation, mais la localisation est d'une grande précision. Cependant capter le signal du satellite dans les bâtiments ou dans les grandes agglomérations, peut y avoir des problèmes de plus le temps de localisation est un peu plus long. Le GPS n'est qu'un système de réception c'est-à-dire qu'il permet juste d'être localisé et non d'envoyer ses coordonnées. Le GPS offre une précision allant de 15 à 100 mètres pour les applications civiles [19].

I.5.1.2. Différents segments du système GPS

Lorsque nous parlons de GPS, nous avons tendance à penser systématiquement et exclusivement à l'appareil qui sert à donner une position. Cependant, le système GPS ne se limite pas à cet instrument, puisqu'il se compose de trois éléments distincts, appelé segments ou secteurs. Le premier segment, composé de satellites est appelé «segment spatial». Le deuxième segment qui comprend des stations de contrôle s'appelle «segment de contrôle». Le dernier correspond aux récepteurs GPS est dit «segment utilisateur».

a- Segment spatial

Le segment spatial est composé d'un réseau de 24 satellites en orbite quasi circulaire autour de la terre à une hauteur à peu près de 20000 km, dont ces satellites sont répartis sur 6 plans orbitaux inclinés 55° par rapport à l'équateur. Cette répartition spatiale garantit la visibilité en permanence d'au moins six satellites en tout point du globe. La durée de vie minimale du système est de 7 ans.



Figure I. 6. Segment spatial.

b- Segment de contrôle

En ce qui concerne le deuxième type de segment, il s'agit d'un réseau de 5 différentes stations de poursuite de la composante de contrôle au sol qui ont pour principale fonction de suivre toutes les secondes la trajectoire des satellites GPS. La station principale gère toutes les transmissions ainsi que le calcul des éphémérides et les corrections des horloges satellite qui sont régulièrement communiquées aux satellites, à leur tour, les transmettent aux utilisateurs. Le récepteur GPS utilise l'ensemble de ces informations pour calculer et corriger sa position. La Figure II.7 illustre les cinq différentes stations de surveillance dispersées dans le monde dont une station maître qui calcule les corrections à apporter aux messages des satellites (Hawaï, Ascension, Diego Garcia, Kwajalein, Colorado Springs).

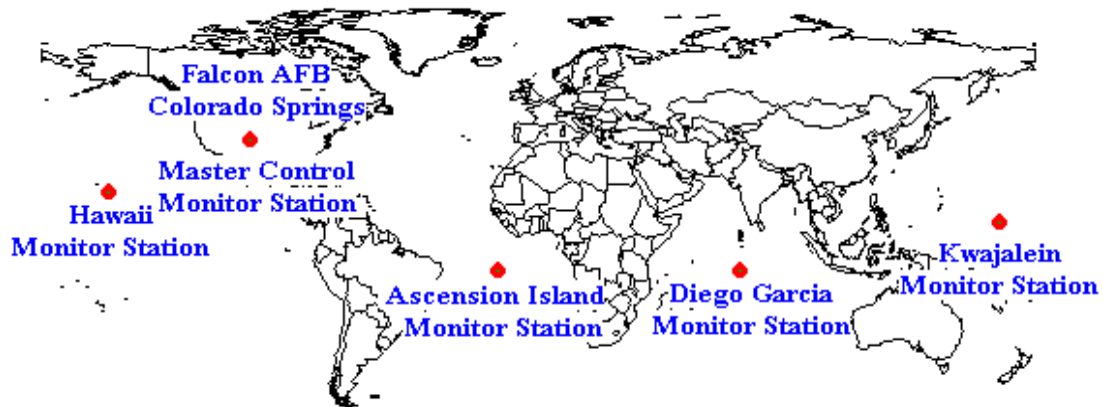


Figure I. 7. Emplacement de segments de Contrôles dans le monde.

Ces stations sont les yeux et les oreilles du système GPS, elles vérifient l'état des satellites lors de leur passage au dessus d'elles. Ces stations transmettent ensuite leurs données à la station principale de Colorado Springs c'est là que les paramètres décrivant l'orbite des satellites, et la qualité des horloges embarquées sont estimés, la vérification de la santé des satellites et la détermination d'un repositionnement éventuel sont également contrôlés. Cette information est alors renvoyée à trois stations de chargement qui transmettent l'information aux satellites grâce à la répartition uniforme des stations de contrôle, tous les satellites GPS sont captés à 92% du temps.

Sur le sol, des stations de contrôle suivent en tout instant le mouvement des satellites, dont l'orbite est périodiquement corrigée [18].

c- Segment utilisateur

Le segment utilisateur est constitué de récepteurs, qui ont été conçus afin de décoder le signal transmis par les satellites, pour déterminer la position la vitesse et le temps. Ceux-ci reçoivent

les signaux des satellites grâce à des récepteurs GPS (voir Figure I.8), qui vont leur permettre de calculer leur position en trois dimensions (latitude, longitude, altitude) et le temps grâce à la triangulation pour comprendre le fonctionnement de ce système, il faut savoir que les ondes émises par les satellites se propagent à la vitesse de la lumière et qu'il est possible de mesurer le temps que met une onde pour parvenir à un récepteurs en comparant l'heure d'émission et de réception de cette dernière. Grâce à ces informations on peut alors calculer la distance qui sépare le satellite du récepteur et ainsi déterminer la position de la cible.

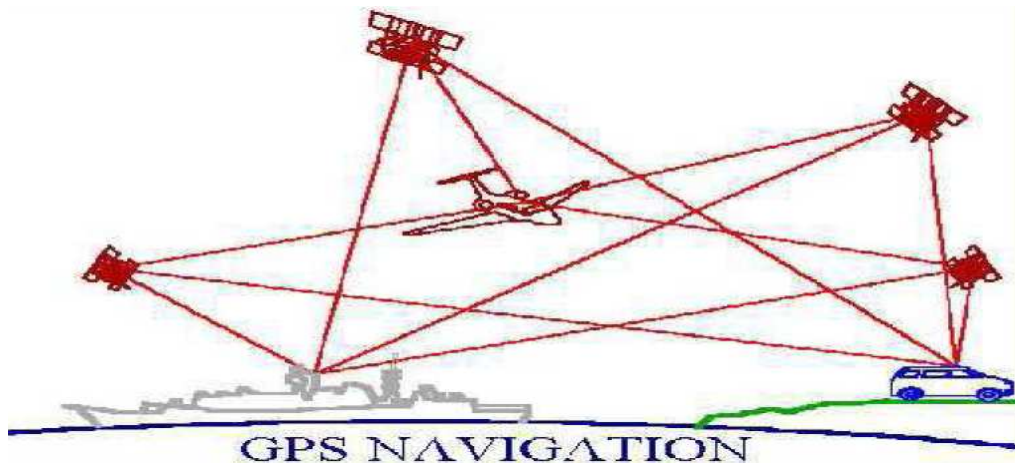


Figure I. 8. Segment utilisateur.

I.5.2. Système GALILEO

À l'inverse du GPS américain et du GLONASS russe, le système GALILEO est développé pour un usage non militaire par l'union européenne pour une couverture globale grâce à 30 satellites (27 opérationnels et 3 en réserve). Ce système est très compétitif au GPS puisqu'il se base sur la même technologie que le GPS mais offre un bien meilleure précision qui va jusqu'à 1 mètre pour le service commercial qui peut être ramené au centimètre en utilisant les corrections des stations de base terrestres avec le système EGNOS déjà déployé. Ce système est non seulement plus précis que les autres systèmes existants mais il est aussi très fiable. Il garanti la validité des données de localisation en intégrant un signal d'intégrité du message. Ce signal d'intégrité permet d'informer l'utilisateur d'une quelconque erreur dans les données reçues.

I.5.3. Système EGNOS

C'est le système prés-GALILEO et, à l'instar du WAAS, le service EGNOS (European Geostationary Navigation Overlay Service) se base sur des stations terrestres, déployées dans plusieurs pays européens. Ces stations permettent de calculer les erreurs de positionnement et d'apporter des corrections via des satellites géostationnaires EGNOS. Ce

système de correction utilise les deux systèmes déjà existants, GPS et GLONASS, afin d'aboutir à une meilleure précision et d'offrir plus de disponibilité à l'utilisateur. Il permet une localisation de moins de 5 mètres et fournit des informations sur l'intégrité des données reçues pour assurer la fiabilité des données de positionnement.

I.5.4. Système MEOSAR

Le système MEOSAR (Medium Earth Orbit Search and Rescue) est conçu pour les opérations de recherche et de sauvetage grâce à des équipements (radiobalises) au sol et à des capteurs qui constituent un second chargement aux satellites qu'on appelle SAR. Ceux-ci se composent d'un récepteur d'ondes à 406Mhz et d'un processeur pour démoduler le signal d'alerte. Les équipements au sol envoient un signal d'alerte sur la bande standard de 406Mhz (accord entre États Unis, Union Européenne et Russie) vers des satellites munis de récepteurs SAR. Les satellites relayent le signal de détresse géo-localisé vers des stations terrestres LUT (Local User Terminal). Les données reçues par les LUT sont transférées vers des centres de traitement qu'on appelle MCC (Mission Control Centers) et, finalement, les données traitées sont aiguillées vers des centres de coordination de sauvetage RCC (Rescue Coordination Centers) qui alertent les services de sauvetage et leur fournissent la position de la balise de détresse activée

I.5.5. Système GLONASS

GLONASS (Système Global de Navigation par Satellite) a été conçu par la Russie lors de la guerre froide pour concurrencer le GPS américain. Il se base sur le système géodésique (calcul des distances spatiales) russe PZ-90.02 conforme au système géodésique mondial WGS84 utilisé par le GPS après une adaptation en septembre 2007 permettant une interopérabilité avec le système GPS. À l'inverse du GPS, les satellites diffusent le même code mais avec des fréquences différentes de chaque satellite. Les satellites utilisent 25 canaux séparés avec un intervalle de 0.5625 MHz en deux plages de fréquence, 1602.5625-1615.5 MHz et 1240-1260 MHz.

Le système compte 23 satellites en 2010, sauf que la durée de vie des satellites est plus limitée par rapport aux satellites GPS. En décembre 2010, le système pourra atteindre une couverture globale de 99.9% avec 24 satellites opérationnels. La modernisation en cours des satellites permettra l'interopérabilité avec GALILEO et GPS, ainsi que l'augmentation de la précision avec la nouvelle génération des satellites GLONASS-K. Le programme GLONASS qui s'étendra jusqu'en 2020, vise à atteindre des performances comparables au GPS et au GALILEO en 2011 [18].

I.5.6. Wifi

Wifi est un ensemble de protocoles de communication sans fil régis par les normes du groupe IEEE 802.11. Grâce aux normes Wifi, il est possible de créer des réseaux locaux sans fil à haut débit. Dans la pratique, le Wifi permet de relier des ordinateurs portables, des machines de bureau, des assistants personnels (PDA : Personale Digital Assistant.), des objets communicants ou même des périphériques à une liaison haut débit (de 11 Mbit/s théoriques ou 6 Mbit/s réels en 802.11b à 54 Mbit/s théoriques ou environ 25 Mbit/s réels en 802.11a ou 802.11g sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) [20].

I.5.6.1. Les usages d'un réseau WiFi

Nous l'avons vu précédemment le wifi est loin d'être la seule solution de réseau hertzien .demeurant depuis plusieurs années la seule technologie ayant un succès commercial notable sur l'étendue elle n'est pas concurrencée. L'Ultra Wide band (UWB) comme le Wimax; auraient pu devenir concurrente du wifi en débordant de leur usage premier; elles n'ont; jusqu'ici; pas tenu leur promesses.

La simplicité d'utilisation et le cout réduit du wifi lui ont permis de conquérir le grand public comme les professionnels.après des débuts difficiles aux USA; puis en Europe en 2001 elle a démocratisé une nouvelle tendance qui est loin d'être éphémère de communications sans fil.

Problématiques et enjeux de GPS et wifi :

-L'utilisation du système et gratuite : seul l'achat du récepteur GPS.

-Le WiFi s'est imposé en indoor.

-Le GPS s'est imposé en outdoor : Limitations dans les environnements où les satellites et le récepteur ne sont pas en visibilité directe avec une précision de 5 à 10m.

-Pas de "GPS en indoor": Limitations due aux multi-trajets [19].

I.5.6.2. Architecture réseau WIFI

Il existe deux modes de fonctionnement :

a) Mode Infrastructure

Le mode infrastructure se base sur une station spéciale appelée Point d'Accès (PA). Ce mode permet à des stations WiFi de se connecter à un réseau (généralement Ethernet) via un point d'accès, comme représenté dans la Figure I.9. Elle permet à une station wifi de se connecter à une autre station wifi via leur PA commun. Une station wifi associée à un autre PA peut aussi s'interconnecter. L'ensemble des stations à portée radio du PA forme un BSS (Basic Service Set). Chaque BSS est identifié par un BSSID (BSS Identifier) de 6 octets qui correspond à

l'adresse MAC du PA. Chaque ordinateur se connecte à un point d'accès via une liaison sans fil. L'ensemble formé par le point d'accès et les stations situées dans sa zone de couverture est appelé ensemble de services de base BSS (basic service set). Il est possible de relier plusieurs BSS (basic service set) par une liaison appelée système de distribution (DS Distribution System) afin de constituer un ensemble de services étendu (Extended service set ou ESS). Le système de distribution (DS) peut être un réseau filaire, ou un câble entre deux points d'accès. Un ESS (Extended Service Set) est repéré par un ESSID (Extended Service Set Identifier), qui est un nom du réseau. Lorsqu'un utilisateur nomade passe d'un BSS (basic service set) à un autre en se déplaçant l'adaptateur réseau sans fil de sa machine est capable de changer de point d'accès selon la qualité des signaux provenant des différents points d'accès. Les points d'accès communiquent entre eux, Cette caractéristique permet aux stations de passer de façon transparente d'un point d'accès à un autre (roaming).



Figure I. 9. Mode infrastructure.

b) Mode Ad-Hoc

La Figure I.10 illustre le fonctionnement de ce mode est totalement distribué, il n'y a pas d'élément structurant hiérarchiquement la cellule ou permettant de transmettre les trames d'une station à une autre. Ce mode permet la communication entre deux machines sans l'aide d'une infrastructure. Les stations se trouvant à portée de radio forment un IBSS (Independent

Basic Service Set). En mode ad hoc les ordinateurs sans fil clients se connectent les uns aux autres pour constituer un réseau point à point (Peer to Peer), c'est un réseau dans lequel chaque ordinateur est un client et un point d'accès. Cet ensemble est appelé, IBSS (indépendant basic service set). Dans un réseau ad hoc, la portée du IBSS est déterminée par la portée de chaque station. Contrairement au mode infrastructure, le mode ad hoc ne propose pas de diffuser régulièrement une trame balise d'une station à une autre. Ainsi un IBSS est par définition comme un réseau sans fil restreint [21].

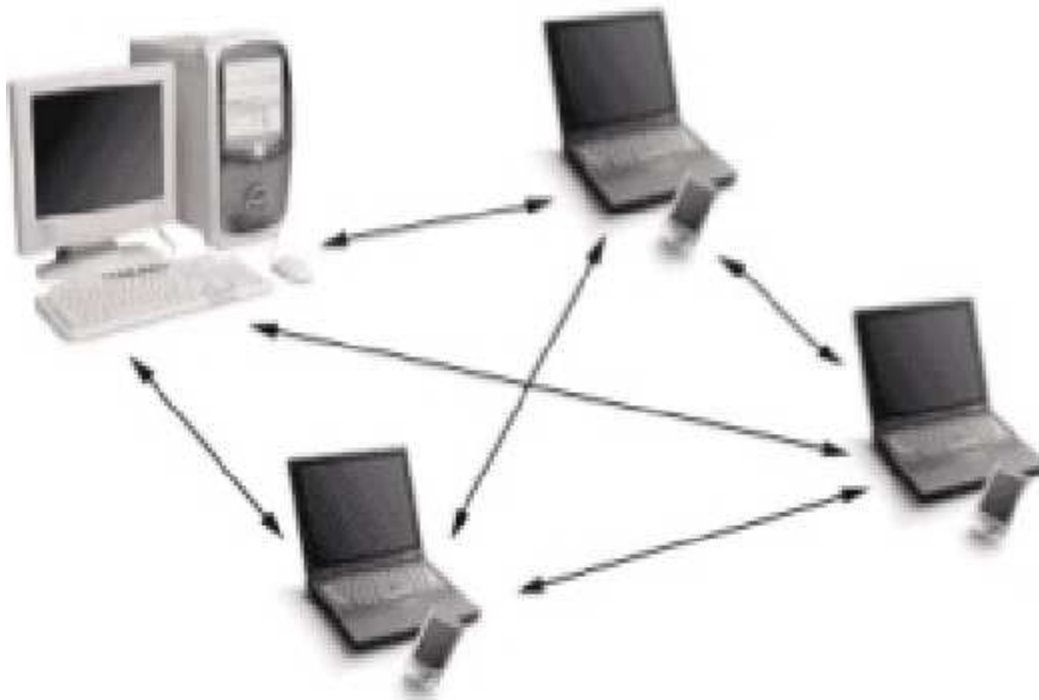


Figure I. 10. Mode Ad-Hoc.

I.6. Conclusion

Nous avons présenté dans ce chapitre une étude bibliographique qui ont introduite les techniques et les technologies de localisation indoor. L'idée est alors de proposer un système de localisation indoor en utilisant la technique WiFi.

Dans le prochain chapitre, nous allons nous intéresser aux techniques de localisation automatique des signaux WiFi en utilisant l'intelligence artificielle, ces dernières techniques sont utilisées pour améliorer les systèmes de localisation indoor.

Chapitre II

*Classification par réseaux de
neurones artificiels et
machines à vecteurs de
support*

II.1. Introduction

Ce chapitre va aborder les différentes notions qui seront utilisées par la suite. Dans un premier temps, une présentation des méthodes de l'intelligence artificielle: les réseaux de neurones artificiels, le réseau de neurones multicouches MLP (fonctions de transfert, architecture de réseau, mise en œuvre des réseaux de neurones) et le réseau à fonction radiale de base RBFN. Ensuite, nous expliquons les techniques de validation les plus utilisées : validation simple, validation croisée et *k-fold* cross-validation. Et nous terminons ce chapitre par les techniques de classification multiclasse.

II.2. Réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones en amont ou des capteurs composant la machine dont il fait partie. A chacune de ses entrées est associé un poids représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones en aval. A chaque connexion est associé un poids. Il est commode de représenter graphiquement un neurone comme indiqué sur la Figure II.1. Cette représentation est à l'origine de la première vague d'intérêt pour les neurones formels, dans les années 1940 à 1970 [22].

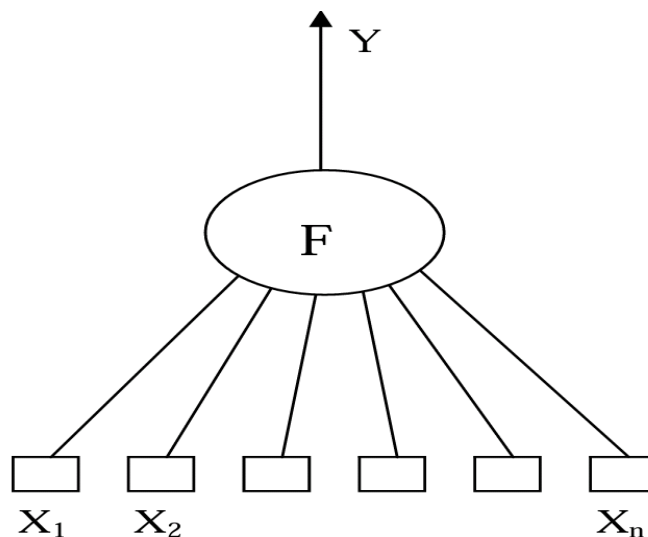


Figure II. 1. Neurone artificiel.

Le neurone réalise alors trois opérations sur ses entrées :

- ✚ **Pondération:** multiplication de chaque entrée par un paramètre appelé poids de connexion,
- ✚ **Sommation:** une sommation des entrées pondérées est effectuée,

✚ **Activation** : passage de cette somme dans une fonction, appelée fonction d'activation. La valeur calculée Y est la sortie du neurone qui est transmise aux neurones suivants.

II.2.1. Réseaux de neurones multicouche MLP

La Figure II.2 montre la structure d'un neurone artificiel. A chacune de ces entrées est associé un poids w abréviation de *weight* (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associé un poids [23].

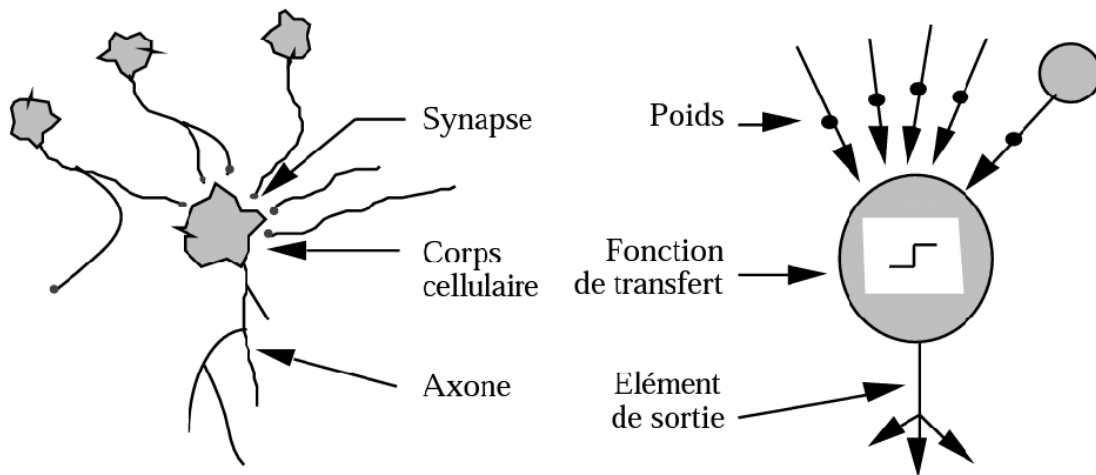


Figure II. 2. Mise en correspondance neurone biologique / neurone artificiel.

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité.

Réseau multicouche (au singulier) : les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales (voir Figure. II.3). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelés couches cachées.

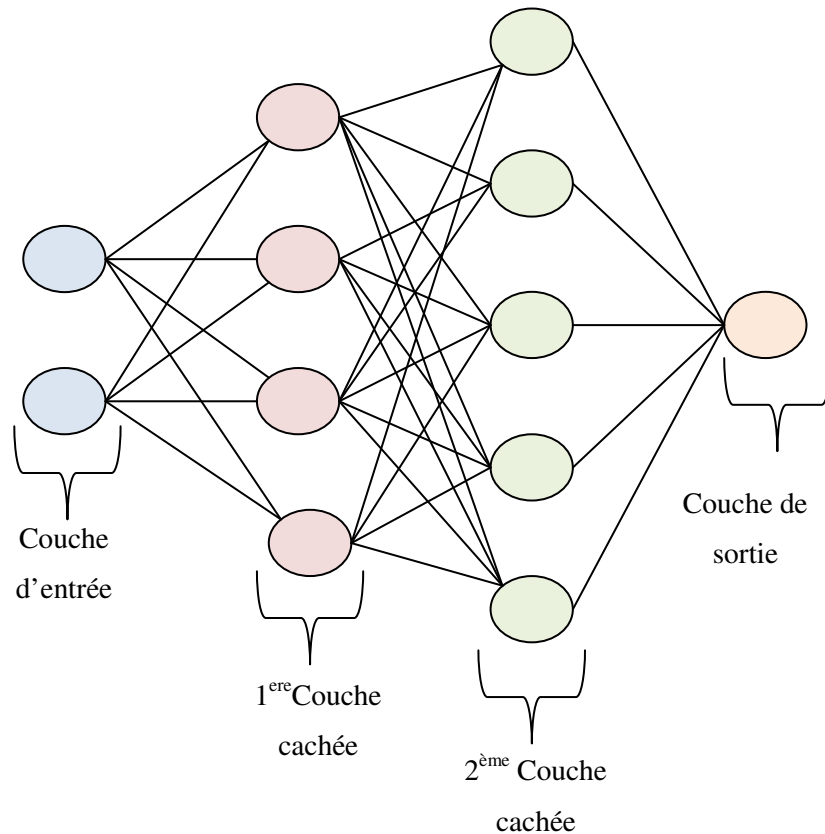


Figure II. 3. Structure d'un réseau de neurone multicouche.

II.2.1.1. Fonctions de transfert

Jusqu'à présent, nous n'avons pas spécifié la nature de la fonction d'activation de notre modèle. Il se trouve que plusieurs possibilités existent. Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone sont énumérées au Tableau II.1.

Les quatre les plus utilisées sont les fonctions "seuil", "linéaire", "sigmoïde", et "tangente hyperbolique" [24].

Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, une entrée négative ne passe pas le seuil, la fonction retourne alors la valeur 0 (on peut interpréter ce 0 comme signifiant faux), alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction retourne 1 (vrai). Utilisée dans le contexte d'un neurone, cette fonction est illustrée à la Figure II.4 a. On remarque alors que le biais b dans l'équation II.1 détermine l'emplacement du seuil sur l'axe n , ou la fonction passe de 0 à 1. Nous verrons plus loin que cette fonction permet de prendre des décisions binaires.

La fonction linéaire est très simple, elle affecte directement son entrée à sa sortie :

$$a = n \quad (\text{II.1})$$

Les fonctions de transferts sigmoïde (Equation II.2) et tangente hyperbolique (Equation II.3) sont données par les équations suivantes :

$$a = \frac{1}{1 + e^{-n}} \quad (\text{II.2})$$

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (\text{II.3})$$

Elle ressemble soit à la fonction seuil, soit à la fonction linéaire, selon que l'on est loin ou près de b (biais), respectivement. La sigmoïde est un compromis intéressant entre les deux précédentes. Notons finalement, que la fonction “tangente hyperbolique” est une version symétrique de la sigmoïde [25].

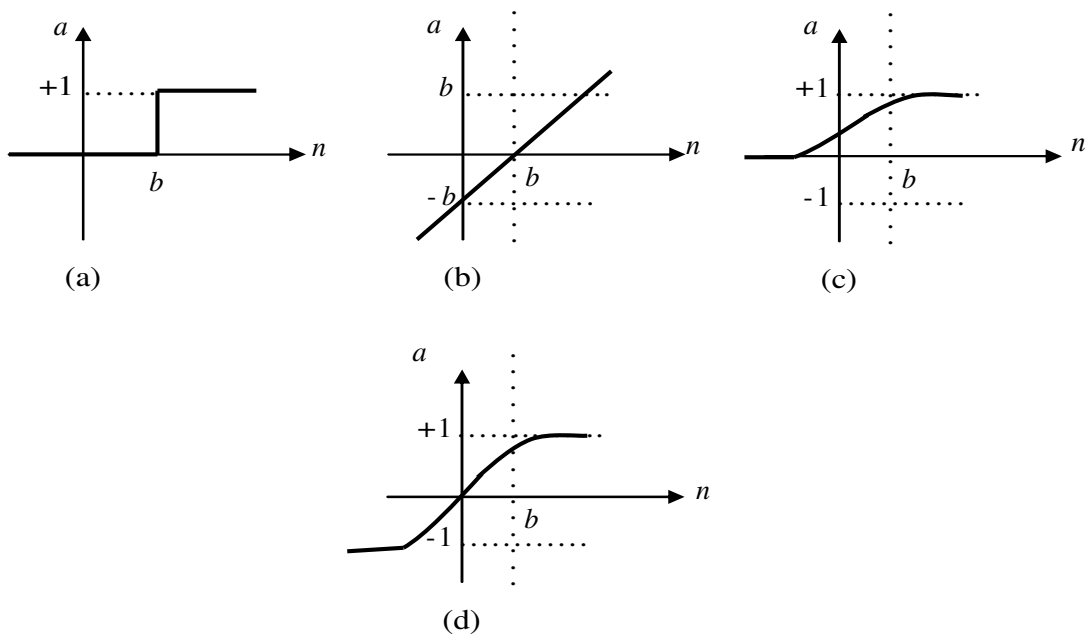
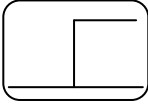
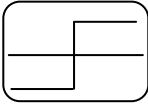
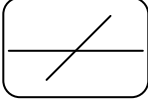
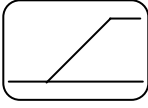
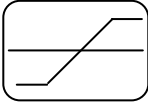
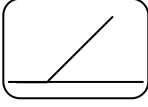
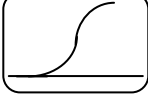
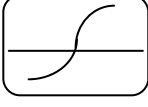



Figure II. 4. Fonction de transfert : (a) du neurone “seuil”; (b) du neurone “linéaire”, (c) “sigmoïde”, et (d) du neurone “tangente hyperbolique”

Le tableau II.1 résume les de transfert couramment utilisée.

Tableau II. 1. Fonctions de transfert $a=f(n)$.

Nom de la fonction	Relation d'entrée / sortie	Icône
Seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
Linéaire	$a = n$	
Linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
Linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
Linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
Sigmoïde	$a = \frac{1}{1 + e^{-n}}$	
Tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Compétitive	$a = 1$ si n maximum $a = 0$ autrement	

II.2.1.2. Architecture de réseau

Un réseau de neurones est un maillage de plusieurs neurones, généralement organisé en couches. Pour construire une couche de S neurones, il s'agit simplement de les assembler comme à la Figure II.5. Les S neurones d'une même couche sont tous branchés aux R entrées. On dit alors que la couche est totalement connectée.

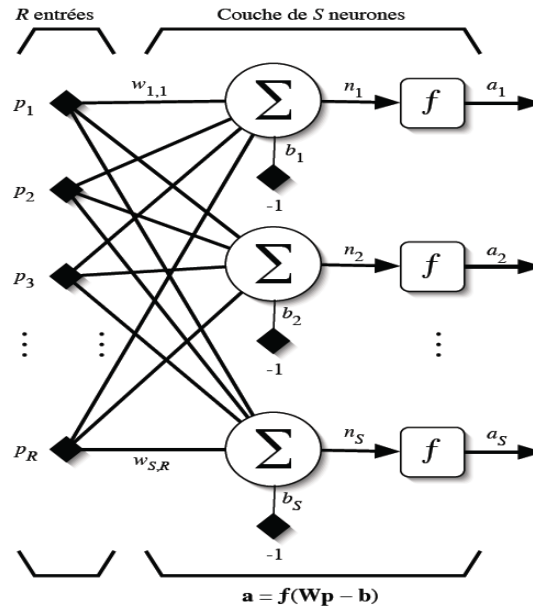


Figure II. 5. Couche de S neurones.

Un poids $w_{i,j}$ est associé à chacune des connexions. Nous noterons toujours le premier indice par i et le deuxième par j . Le premier indice (rangée) désigne toujours le numéro de neurone sur la couche, alors que le deuxième indice (colonne) spécifie le numéro de l'entrée. Ainsi, $w_{i,j}$ désigne le poids de la connexion qui relie le neurone i à son entrée j . L'ensemble des poids d'une couche forme donc une matrice W de dimension $S \times R$:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \vdots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix} \quad (\text{II.4})$$

Notez bien que $S \neq R$, dans le cas général (les nombres de neurones et d'entrées sont indépendants). Si l'on considère que les S neurones forment un vecteur de neurones, alors on peut créer les vecteurs $\mathbf{b} = [b_1, b_2, \dots, b_S]^T$, $\mathbf{n} = [n_1, n_2, \dots, n_S]^T$ et $\mathbf{a} = [a_1, a_2, \dots, a_S]^T$. Ceci nous amène à la représentation graphique simplifiée, illustrée à la Figure II.6. On y retrouve,

comme à la Figure II.5 les mêmes vecteurs et matrice. La seule différence se situe au niveau de la taille, ou plus précisément du nombre de rangées (S), de b , n , a et W .

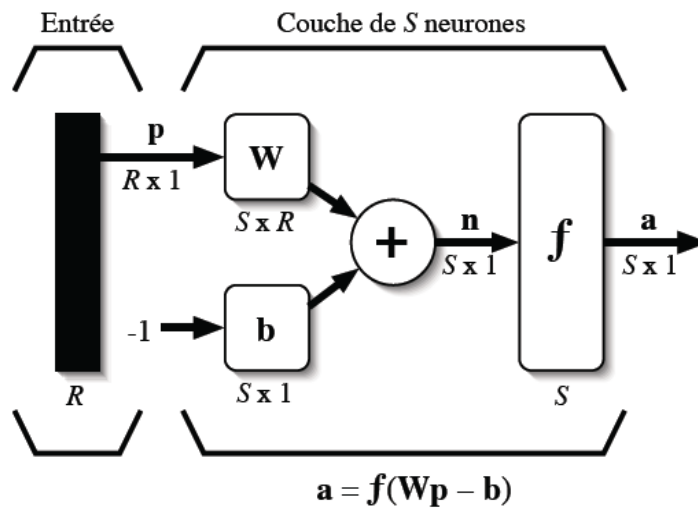


Figure II. 6. Représentation matricielle d'une couche de S neurones.

Finalement, pour construire un réseau, il ne suffit plus que de combiner des couches comme à la Figure II.7. Cet exemple comporte R entrées et trois couches de neurones comptant respectivement S^1 , S^2 et S^3 neurones. Dans le cas général, de nouveau, S^1 , S^2 , S^3 , chaque couche possède sa propre matrice de poids W^k , où k désigne l'indice de couche. Dans le contexte des vecteurs et des matrices relatives à une couche, nous emploierons toujours un exposant pour désigner cet indice. Ainsi, les vecteurs b^k , n^k et a^k sont aussi associés à la couche k .

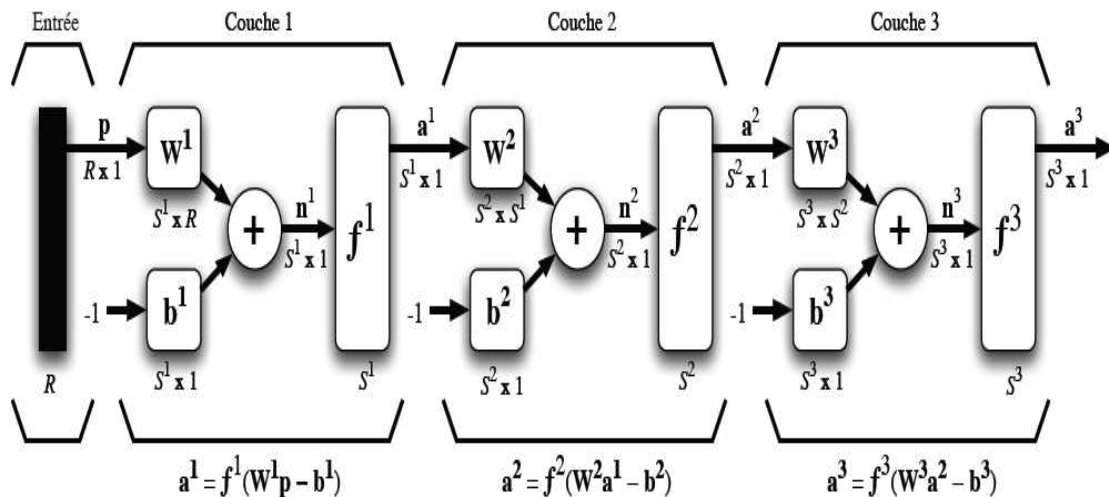


Figure II. 7. Représentation matricielle d'un réseau de trois couches.

Il est important de remarquer dans cet exemple que les couches qui suivent la première ont comme entrée la sortie de la couche précédente. Ainsi, on peut enfileur autant de couche que l'on veut, du moins en théorie. Nous pouvons aussi fixer un nombre quelconque de neurones

sur chaque couche. En pratique, nous verrons plus tard qu'il n'est cependant pas souhaitable d'utiliser trop de neurones. Finalement, notez aussi que l'on peut changer de fonction de transfert d'une couche à l'autre. Ainsi, toujours dans le cas général $f^1 \neq f^2 \neq f^3$.

La dernière couche est nommée «couche de sortie». Les couches qui précèdent la couche de sortie sont nommées «couches cachées». Nous verrons un peu plus tard pourquoi. Le réseau de la Figure II.7 possède donc deux couches cachées et une couche de sortie.

Les réseaux multicouches sont beaucoup plus puissants que les réseaux simples à une seule couche. En utilisant deux couches (une couche cachée et une couche de sortie), à condition d'employer une fonction d'activation sigmoïde sur la couche cachée, on peut entraîner un réseau à produire une approximation de la plupart des fonctions, avec une précision arbitraire (cela peut cependant requérir un grand nombre de neurones sur la couche cachée). Sauf dans de rares cas, les réseaux de neurones artificiels exploitent deux ou trois couches.

Entraîner un réseau de neurones signifie modifier la valeur de ses poids et de ses biais pour qu'il réalise la fonction entrée/sortie désirée. Nous étudierons en détails, dans les sous sections suivantes, la mise en œuvre des réseaux de neurones et l'algorithme d'apprentissage.

II.2.1.3. Mise en œuvre des réseaux neuronaux

Nous allons suivre une démarche reprise par Wierenga et Kluytmans [26] qui est composée de quatre étapes principales :

Étape 1: Fixer le nombre de couches cachées

Mis à part les couches d'entrée et de sortie, l'analyste doit décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation; avec une couche cachée, il est capable, avec un nombre suffisant de neurones, d'approximer toute fonction continue [27]. Une seconde couche cachée prend en compte les discontinuités éventuelles.

Étape 2: Déterminer le nombre de neurones par couches cachées

Chaque neurone supplémentaire permet de prendre en compte des profils spécifiques des neurones d'entrée. Un nombre plus important permet donc de mieux coller aux données présentées mais diminue la capacité de généralisation du réseau. Ici non plus il n'existe pas de règle générale mais des règles empiriques.

Étape 3: Choisir la fonction d'activation

Nous considérerons la fonction logistique pour le passage de la couche d'entrée à la couche cachée. Le passage de cette dernière à la couche de sortie sera soit linéaire, soit sigmoïde (logistique), soit tangente hyperbolique, selon nos types de variables.

Étape 4: Choisir l'apprentissage

L'apprentissage par rétro-propagation nécessite la détermination du paramètre d'ajustement des poids synaptiques à chaque itération.

La détermination du critère d'arrêt est aussi cruciale dans la mesure où la convergence peut passer par des minima locaux [28].

II.2.1.4. Algorithmes apprentissage

Supposons que l'on dispose d'une base d'apprentissage de taille k d'observations (x_1, x_2, \dots, x_k) et de la variable à prévoir y_i . Considérons le cas le plus simple de la régression avec un réseau constitué d'un neurone de sortie linéaire et d'une couche à n neurones dont les paramètres sont optimisés par moindres carrés. Ceci se généralise à toute fonction perte dérivable et donc à la discrimination à m classes. La prédiction de notre modèle neuronal est donnée par \hat{y}_i .

On considère un réseau comportant une couche d'entrée à n neurones, une couche de sortie à m neurones et il comporte une à plusieurs couches cachées.

Supposons qu'on dispose d'un ensemble d'apprentissage composé de k paires de vecteurs :

$$(x_1, o_1), (x_2, o_2), \dots, (x_k, o_k)$$

avec :

$$\begin{aligned} x_p &= (x_{p,0,1}, x_{p,0,2}, \dots, x_{p,0,n})^t \in R^n && \text{Vecteur d'entrée.} \\ O_p &= (O_{p,1}, O_{p,2}, \dots, O_{p,m})^t \in R^m && \text{Vecteur des sorties désiré.} \\ y_p &= (y_{p,L,1}, y_{p,L,2}, \dots, y_{p,L,n})^t \in R^n && \text{Vecteur des sorties réel du réseau.} \end{aligned}$$

L'apprentissage est déterminé par la minimisation de la fonction perte quadratique ou de celle d'une fonction d'erreur en classification sur le $k^{\text{ème}}$ nœud de sortie est [29]:

$$\delta_{pk} = O_{pk} - x_{pk} \quad (\text{II.5})$$

Dans la littérature scientifique, différents algorithmes d'optimisation sont proposés, ils sont généralement basés sur une évaluation du gradient par rétro-propagation [30].

Par conséquent l'erreur totale (pour tous les nœuds) est [29]:

$$E_p = \frac{1}{2} \sum_{k=1}^m \delta_{p,k}^2 = \frac{1}{2} \sum_{k=1}^m (O_{p,k} - x_{p,l,k})^2 \quad (\text{II.6})$$

Pour minimiser E_p (erreur quadratique), on calcule son gradient par rapport à chaque poids w , puis on modifie les poids dans le sens inverse du gradient.

La fonction de mise à jour des poids est en fonction du calcul du gradient. Ainsi, les poids sur la couche des sorties sont mis à jour de la façon suivante [29]:

$$w_{l,k,j}(t+1) = w_{l,k,j}(t) + \Delta_p w_{l,k,j}(t) \quad (\text{II.7})$$

$$\Delta_p w_{l,k,j}(t) = \mu(O_{p,k} - x_{p,l,k}) f'(y_{p,l,k}) x_{p,l-1,j} \quad (\text{II.8})$$

ou : μ est le pas (taux) d'apprentissage.

La convergence du réseau par rétro-propagation est un problème crucial car il requiert de nombreuses itérations. Pour pallier à ce problème, un paramètre est souvent rajouté pour accélérer la convergence. Ce paramètre est appelé le momentum.

Le momentum est un moyen efficace pour accélérer l'apprentissage et aussi pour pouvoir sortir des minimums locaux.

La règle de mise à jour des poids devient alors :

$$w_{l,k,j}(t+1) = w_{l,k,j}(t) + \mu e_{p,l,k} x_{p,l-1,j} + m[w_{l,k,j}(t) - w_{l,k,j}(t-1)] \quad (\text{II.9})$$

avec $e_{p,l,k} = \mu(O_{p,k} - x_{p,l,k}) f'(y_{p,l,k}) x_{p,l-1,j}$

m : est la constante du momentum.

Généralement, le pas d'apprentissage μ et le momentum m doit être compris entre 0 et 1 pour assurer la convergence de l'algorithme vers une solution optimale.

Le résumé l'algorithme de Rétro-propagation est donnée par les étapes suivantes :

- Étape (1):** Appliquer un vecteur d'entrée $x_p = (x_{p,0,1}, x_{p,0,2}, \dots, x_{p,0,n})^t$ aux nœuds d'entrées puis initialiser les poids du réseau à de petites valeurs aléatoires;
- Étape (2):** Exécuter l'échantillon d'apprentissage à travers le réseau;
- Étape (3):** Calculer les termes d'erreur de signal de la couche de sortie et les couches cachées;
- Étape (4):** Mise à jour les poids de la couche de sortie et couches cachées;
- Étape (5):** Répéter ce processus jusqu'à ce que l'erreur E_p devienne acceptable (aller à l'étape 2).

Il n'existe pas de règles permettant de déterminer le nombre de couches cachées dans un réseau donné n_i , le nombre de neurones dans chacune d'elles, le pas d'apprentissage μ , et le momentum m . La solution est d'optimiser ces paramètres d'une façon empiriques.

II.2.2. Réseaux à fonction radiale de base RBFN

Les réseaux à fonctions radiale de base, en anglais Radial Basis Function Network (RBFN), sont des réseaux similaires aux MLP en termes d'architectures et d'algorithme d'apprentissage. La différence réside dans l'utilisation des fonctions de transfert à base radiale (gaussienne) [31].

II.2.2.1 Architecture de base

La Figure II.8 montre l'architecture générale d'un RBFN. Dans les neurones de la couche cachée sont généralement utilisées des fonctions de transfert gaussiennes qui jouent le rôle de fonction noyau de modèle. Dans les neurones de la couche d'entrée et la couche de sortie, le plus fréquent est d'utiliser les fonctions de transfert linéaires.

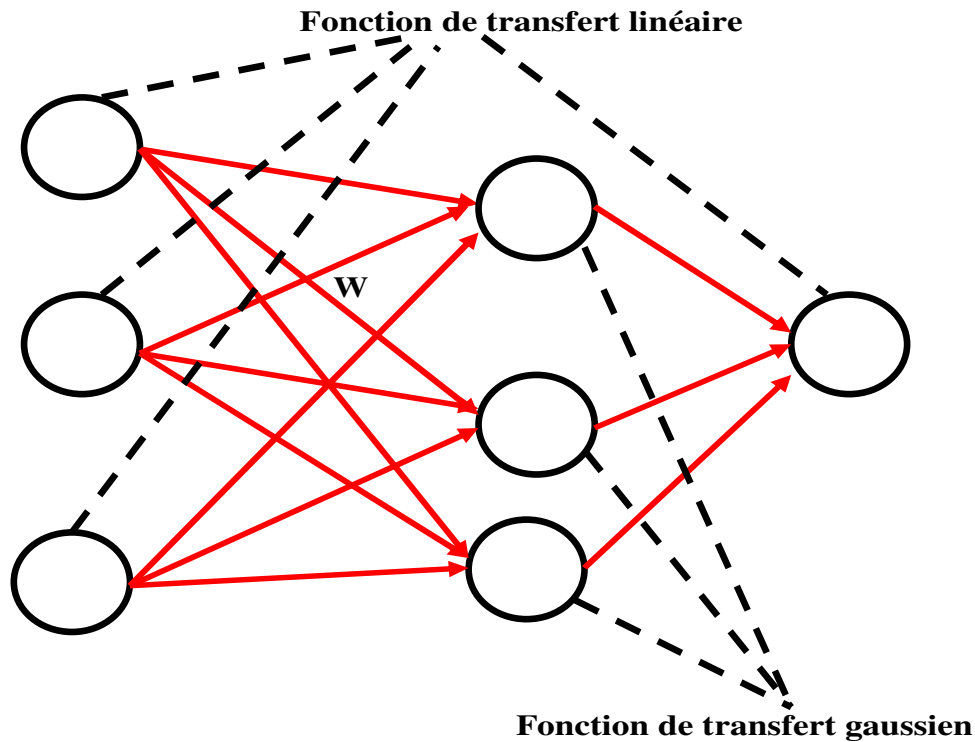


Figure II. 8. Architecture générique d'un RBFN.

II.2.2.2. Caractéristiques de base

Tout comme les MLP, les RBFN possèdent eux aussi la propriété d'approximation universelle [32]. Les entrées sont propagées aux noyaux gaussiens Φ_j avec $j = \{1, 2, \dots, m\}$, m étant le nombre de neurones cachés. Les valeurs du noyau sont ensuite pondérées par des coefficients λ_j afin de produire la valeur de sortie \hat{y} , ou \hat{y} est une approximation de la sortie désirée y .

$$\hat{y} = \sum_{j=1}^m \lambda_j \Phi_j(X, c_j, \sigma_j) \quad (\text{II.10})$$

Les noyaux gaussiens sont définis par :

$$\Phi_j(X, c_j, \sigma_j) = \exp\left(-\frac{\|X - c_j\|^2}{2\sigma_j}\right) \quad (\text{II.11})$$

X est le variable d'entrée et les paramètres (c_j, σ_j) désignent respectivement le centre et la largeur de la fonction gaussienne. L'intérêt principal provient du fait que les paramètres peuvent être appris indépendamment, avec un apprentissage relativement rapide pour les λ_j comme solution d'un système d'équations linéaires. On peut trouver les détails de l'apprentissage du RBFN dans [8] et [33]. En pratique, les paramètres c_j influencent de façon relativement limitée le résultat final, alors que les paramètres σ_j améliorent sensiblement l'approximation de la relation entrées-sortie. Théoriquement, il faut que les noyaux gaussiens soient assez larges et se recouvrent suffisamment pour que les parties de la distribution où aucun noyau n'est venu se placer soient prises en compte par le RBFN. En même temps, ils ne doivent pas être trop larges pour qu'ils ne recouvrent pas l'ensemble de la distribution. Les paramètres λ_j sont uniques, une fois que c_j et σ_j sont fixés. Leur influence sur la sortie est directe.

L'architecture d'un réseau RBFN s'organise en deux couches seulement: une couche cachée et une couche de sortie. La première couche, constituée de j noyaux élémentaires, effectue une transformation non linéaire de l'espace d'entrée. La couche de sortie calcule une combinaison linéaire des sorties des noyaux élémentaires.

Dans notre travail, les paramètres à optimiser dans le cette cas sont : nombre de neurones dans la couche cachée (NN) et la largeur de gaussien σ_j , ces paramètres sont optimisés de façon empirique avec la technique validation simple.

II.2.2.3. Variantes du RBFN

On trouve dans la littérature de nombreuses applications directes des RBFN pour la prévision et la classification dans différents domaines [8]. Par ailleurs, il est tout à fait possible de combiner de tels réseaux avec les modèles statistiques de séries temporelles comme dans le cas des MLP. Dans [34], le RBFN donne en sortie la valeur de prédiction mais aussi une valeur de confiance (classe). Les valeurs à faible confiance sont alors ignorées.

Dans cette étude, nous nous sommes limités aux réseaux de neurones de type MLP et RBFN. Ils se prêtent le mieux à notre application. De plus, la relative facilité avec laquelle on peut analyser leur fonctionnement dans le cadre de la localisation de l'environnement intérieur avec l'utilisation de signal WiFi.

Les paramètres ajustables dans un réseau RBFN tel que présenté par la Figure II.8 sont :

- La position des centres c_j ($1 \leq j \leq NN$).
- L'optimisation du nombre NN de noyaux.

- La valeur de l'écart-type associé à chaque noyau.
- Les poids de la couche de sortie W .

Les avantages suivants du RBFN par rapport au MLP avec l'algorithme de rétro-propagation ont été expérimentalement et théoriquement avérés:

- L'apprentissage dans RBFN est plus rapide que l'apprentissage tel que le MLP.
- Une meilleure généralisation est réalisée dans RBFN.
- RBFN ont les propriétés très rapides de convergence comparées aux réseaux multicouche conventionnels avec des fonctions de transfert sigmoïdes.
- Il n'y a aucun problème de minimum local.
- La couche cachée a une interprétation beaucoup plus claire que le MLP avec l'algorithme de rétro-propagation. Il est plus facile d'expliquer ce qu'un réseau de RBFN a appris que ses contre parties MLP avec l'algorithme de rétro-propagation.

Il y a également des inconvénients à employer le RBFN, l'un d'entre eux trouvant le nombre approprié de nœuds cachés. L'étude non surveillée pourrait être nécessaire pour s'appliquer d'abord et pour découvrir le nombre de faisceaux. Le nombre de nœuds cachés est alors placé pour être égal à ce nombre. Un trop grand nombre, ou un trop petit nombre des nœuds cachés empêchera RBFN d'approximer correctement les données [35].

II.3. Machine à vecteurs de support

Les machines à vecteurs de support (SVM Support Vector Machines) sont des nouvelles techniques discriminantes dans la théorie de l'apprentissage statistique. Elles ont été proposées en 1995 par V. Vapnik dans son livre *The nature of Statistical Learning Theory* [36]. Elles permettent d'aborder plusieurs problèmes divers et variés comme la régression, la classification, la fusion etc [10, 33, 37, 38]. Les SVM ont été développé comme un Algorithme de classification binaire supervisée. Il s'avère particulièrement efficace de par le fait qu'il peut traiter des problèmes mettant en jeu de grands nombres de descripteurs, qu'il assure une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones MLP) et il a fourni de bons résultats sur des problèmes réels. L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre espace permettant d'augmenter la séparabilité des données. On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initiale [39].

II.3.1. Principe général

Cette technique est une méthode de classification à deux classes qui tente de séparer les exemples positifs et les exemples négatifs dans l'ensemble de donnée. La méthode cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Cela garantit une généralisation du principe car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être situés d'un côté ou l'autre de la frontière. L'intérêt de cette méthode est la sélection de vecteurs supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas, ce qui peut être considéré comme un avantage pour cette méthode.

On se limite pour l'instant à un problème de discrimination à deux classes, c'est-à-dire $y \in \{1, -1\}$, le vecteur d'entrée x étant dans un espace X muni d'un produit scalaire. On peut prendre par exemple $X \in \mathbb{R}^N$.

II.3.2. Notions de base: Hyperplan, marge et support vecteur

Pour deux classes d'exemples donnés (voir la Figure II.9), le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan. Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.

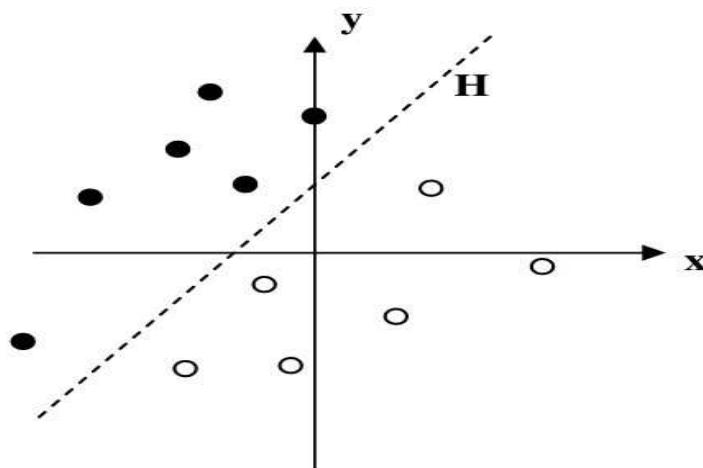


Figure II. 9. Exemple d'un hyperplan séparateur.

Comme représenté la Figure II.10, les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

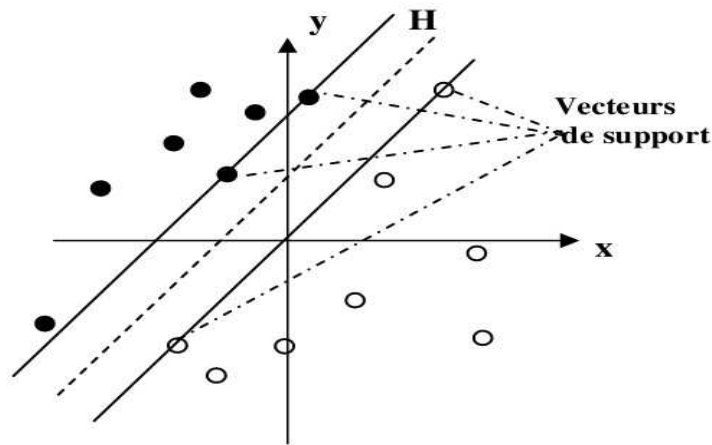


Figure II. 10. Exemple de vecteurs de support.

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « Plus sûr ».

En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale.

On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [39].

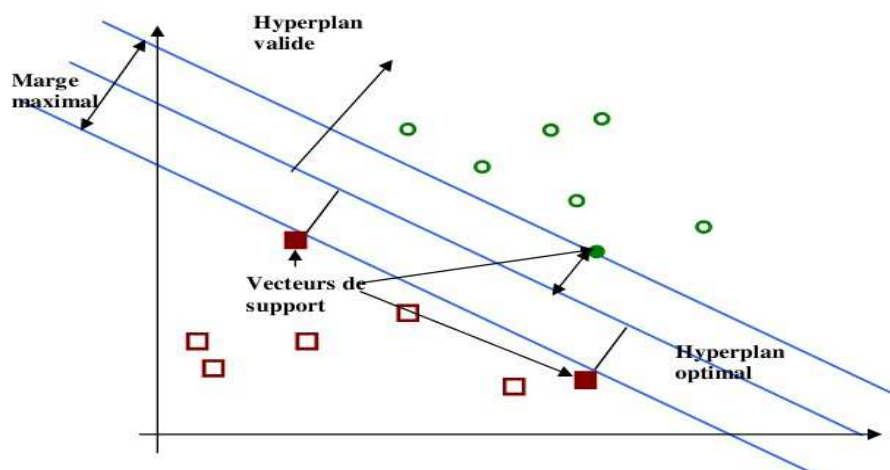


Figure II. 11. Exemple de marge maximale (hyperplan valide).

S'il existe un hyperplan qui sépare les deux classes, les points de l'hyperplan sont décrits par l'équation :

$$x_i \omega + b = 0 \quad (\text{II.12})$$

Où ω est la normale au plan et $|b|/\|\omega\|$ la distance entre l'hyperplan et l'origine, comme représenté dans la Figure II.11.

Pour estimer la classe d'un nouveau point x , on calcule [40]:

$$f(x) = \text{sign}(x \cdot \omega^0 + b^0) = \text{sign}\left(\sum_{VS} \alpha_i^0 y_i(x_i, x) + b^0\right) \quad (\text{II.13})$$

où ω^0, b^0 sont des constantes qui vérifiant l'hyperplan optimal, VS : est l'ensemble des vecteurs de support $0 \leq \alpha_i^0 \leq C$.

Les α_i^0 sont les multiplicateurs de Lagrange. Le paramètre de régularisation C contrôle le niveau d'erreur dans la classification (sert à fixer le compromis entre la minimisation de l'erreur d'apprentissage et la maximisation de la marge).

Pour généraliser cette méthode dans le cas où la fonction de décision n'est pas linéaire, l'idée est la transformation vers un autre espace de dimension suffisamment grande en utilisant une fonction Φ , tel qu'il existe une fonction \mathbf{K} de noyau. Le passage d'un espace vers un autre espace à hyper dimension est donné par l'équation II.14, et représenté dans la Figure II.12.

$$\mathbf{K}(x_i, x) = \Phi(x_i)^T \cdot \Phi(x) \quad (\text{II.14})$$

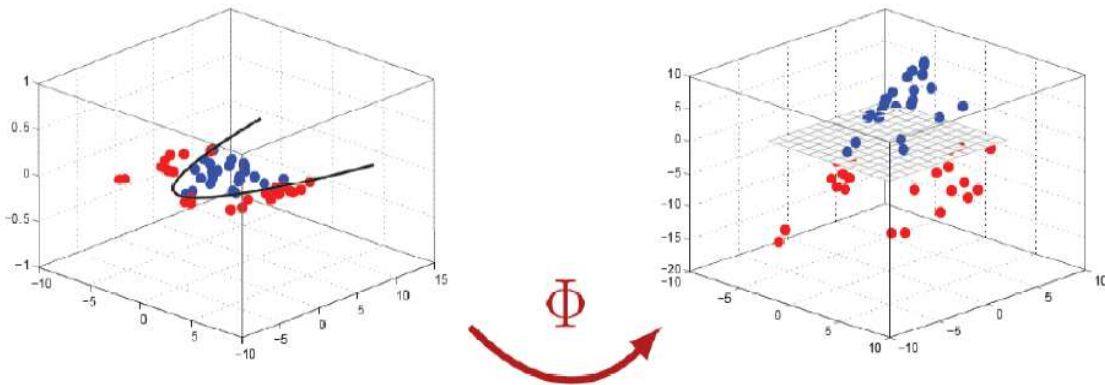


Figure II. 12. Transformer l'espace de départ vers un espace de dimension supérieure.

Pour estimer la classe d'un point x , il suffit de calculer la fonction [41]:

$$f(x) = \text{sign}\left(\sum_{VS} \alpha_i^0 y_i \mathbf{K}(x_i, x) + b^0\right) \quad (\text{II.15})$$

On distingue deux cas de séparation, le cas linéairement séparable et le cas non linéairement séparable (voir la Figure II.13). La première est les plus simples de SVM car il permet de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de

séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

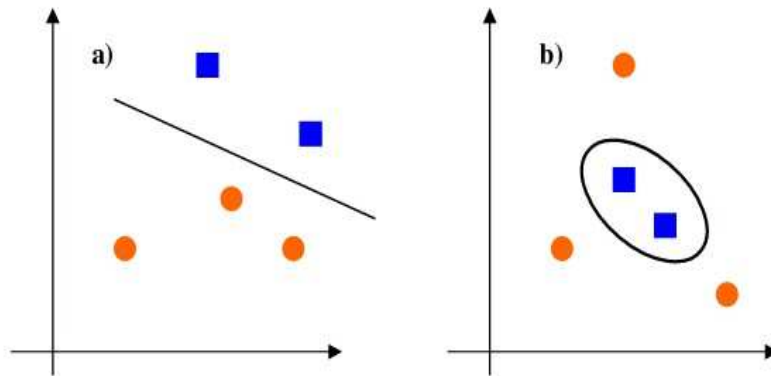


Figure II. 13. a) Cas linéairement séparable, b) Cas non linéairement séparable.

Pour choisir le noyau \mathbf{K} : les principaux noyaux utilisés dans les applications sont [42, 43]:

➤ **Le noyau RBF Gaussien:**

$$\mathbf{K}(x_i, x) = \exp\left(-\gamma \|x_i - x\|^2\right) \quad (\text{II.16})$$

γ : Paramètre inversement proportionnel à la largeur de la gaussienne (σ), avec $\gamma = \frac{1}{2\sigma^2}$.

➤ **Le noyau Polynomial de degré d :**

$$\mathbf{K}(x_i, x) = (x_i^T \cdot x + 1)^d \quad (\text{II.17})$$

➤ **Le noyau Linéaire:**

$$\mathbf{K}(x_i, x) = (x_i^T \cdot x) \quad (\text{II.18})$$

Nous allons donc appliquer cette approche pour la classification des panneaux de signalisation. Dans notre travail, nous avons choisi le noyau RBF, donc les paramètres à optimiser dans ce cas sont : C et γ , ces paramètres sont optimisés de façon empirique avec la technique validation simple.

Les avantages de SVM sont [44]:

- Capacité à traiter de grandes dimensionnalités.
- Traitement des problèmes non linéaires avec le choix des noyaux.
- Souvent performant dans les comparaisons avec les autres approches.

Les inconvénients de SVM sont [44]:

- Difficulté à identifier les bonnes valeurs des paramètres (et sensibilité aux paramètres).

- Le traitement des problèmes multi-classes reste une question ouverte.
- Problème lorsque les classes sont bruitées (multiplication des points supports).
- Pas de modèle explicite pour les noyaux non linéaires (utilisation des points supports).

II.4. Technique de validation

Dans la littérature scientifique, il existe plusieurs techniques pour l'optimisation des meilleurs paramètres de l'algorithme de classification (MLP, RBFN, SVM) [45]:

- Validation simple,
- Monte-Carlo validation
- *k-fold* cross-validation

II.4.1. Validation simple

La technique de validation simple consiste tout simplement à diviser les données disponibles en deux ensembles (les ensembles d'apprentissage et de validation), sans qu'une donnée ne soit commune, un nombre non négligeable de données est bien entendu nécessaire dans l'ensemble de validation pour estimer correctement le critère de généralisation, donc pour optimiser les paramètres de classifieur (MLP, RBFN ou SVM). Souvent on garde 2/3 des données dans l'ensemble d'apprentissage et en réserver 1/3 pour la validation. Les ensembles d'apprentissage et de validation sont tirés aléatoirement parmi les N données disponibles.

II.4.2. Monte-Carlo validation Croisé

L'inconvénient majeur de la technique de validation simple est que ses résultats dépendent intimement du tirage aléatoire des données, entre les deux ensembles d'apprentissage et de validation. Un remède possible consiste alors à effectuer plusieurs tirages de ce type, et estimer l'erreur de généralisation pour chacun de ces tirages, et à retenir la moyenne de ces estimations [46].

II.4.3. *k-fold* cross-validation

Si l'augmentation du nombre de tirages (et donc du nombre d'évaluations dans la méthode de Monte-Carlo validation croisé) permet d'utiliser successivement les mêmes données, tantôt pour l'apprentissage, tantôt pour la validation, leur caractère aléatoire empêche toute certitude quant au fait que chaque donnée soit, au moins approximativement, utilisée le même nombre de fois pour l'apprentissage ou pour la validation. Seule une valeur prohibitive de J permettrait d'atteindre cette certitude.

Une variante consiste alors à découper l'ensemble des N données disponibles en k sous-ensembles disjoints de taille approximativement identiques. On peut alors obtenir k évaluations, en utilisant à chaque fois $k-1$ sous-ensembles pour l'apprentissage et un pour la validation. Chacun des k sous-ensemble joue alors une et une seule fois le rôle d'ensemble de

validation. L'estimation de l'erreur de généralisation est alors calculée par la moyenne des erreurs de généralisation de k évaluations.

Dans notre cas, nous nous sommes intéressés à la technique de k -fold cross-validation, avec k égal à 5.

II.5. Classification multiclasse

Les machines à vecteur support sont dans leur origine binaire. Cependant, les problèmes du monde réel sont dans la plupart des cas multiclasse, l'exemple le plus simple en est la classification de l'environnement avec l'utilisation de la puissance du signal WiFi comme entrée de notre modèle d'apprentissage. Dans de tels cas, on ne cherche pas à affecter un nouvel exemple à l'une de deux classes mais à l'une parmi plusieurs, c-à-d que la décision n'est plus binaire et un seul hyperplan ne suffit plus.

Les méthodes des machines à vecteur support multiclasse, réduisent le problème multiclasse à une composition de plusieurs hyperplans biclasses permettant de tracer les frontières de décision entre les différentes classes. Ces méthodes décomposent l'ensemble d'exemples en plusieurs sous ensembles représentant chacun un problème de classification binaire. Pour chaque problème un hyperplan de séparation est déterminé par la méthode SVM binaire (MLP, RBF). On construit lors de la classification une hiérarchie des hyperplans binaires qui est parcourue de la racine jusqu'à une feuille pour décider de la classe d'un nouvel exemple. On trouve dans la littérature plusieurs méthodes de décomposition: une contre reste, une contre une, graph de décision, arbres de décision par la méthode SVM [47]. Dans notre travail nous intéressent à la méthode une contre reste (1vsR).

II.5.1. Une-contre-une (1vs1)

Cette méthode, appelée aussi "pairwise", revient à Kner et ses co-auteurs [48] qui l'ont proposée pour les réseaux de neurones. Elle consiste à utiliser un classifieur pour chaque paire de classes. Au lieu d'apprendre K fonctions de décisions, la méthode 1vs1 discrimine chaque classe de chaque autre classe, ainsi $K(K-1)/2$ fonctions de décisions sont apprises. Pour chaque paire de k contre s classe, la méthode une-contre-une définit une fonction de décision biclasses $h_{ks} : R \rightarrow \{-1, +1\}$. L'affectation d'un nouvel exemple se fait par liste de vote. On teste un exemple par le calcul de sa fonction de décision pour chaque hyperplan. Pour chaque test, on vote pour la classe à laquelle appartient l'exemple (classe gagnante). On définit la fonction de décision biclasses $H_{ks}(x)$ par l'équation suivante :

$$H_{ks}(x) = \text{signe}(f_{ks}(x)) \quad (\text{II.19})$$

$$H_{ks}(x) = \begin{cases} +1 & \text{si } f_{ks}(x) > 0 \\ 0 & \text{sinon} \end{cases} \quad (\text{II.20})$$

Sur la base des $K(K-1)/2$ fonctions de décision biclasses, on définit K autres fonctions de décision :

$$H_k(x) = \sum_{s=1}^m H_{ks}(x) \quad (\text{II.21})$$

Un nouvel exemple est affecté à la classe la plus votée. La règle de classification d'un nouvel exemple x est donnée par l'équation :

$$k^* = \arg_{1 \leq k \leq K} \text{Max}(H_k(x)) \quad (\text{II.22})$$

Malheureusement, l'équation II.22 peut être vérifiée pour plusieurs classes, ce qui produit des zones d'indécisions. La méthode de vote affecte dans ce cas, un exemple aléatoirement à l'une des classes les plus votées.

La Figure II.14 représente un exemple de classification de trois classes avec la zone d'indécision.

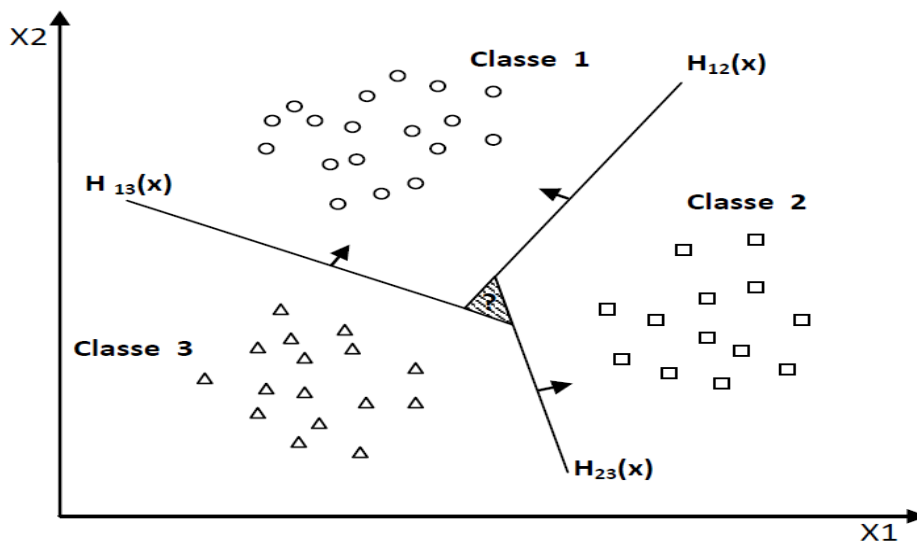


Figure II. 14. Approche une-contre-une.

Bien que La méthode une-contre-une utilise, pour l'entraînement, un nombre plus important d'hyperplans que la méthode une-contre reste, elle est souvent plus rapide. Cela est du, d'une part, au nombre limité d'exemples utilisés pour entrainer chaque hyperplan, et d'autre part, à la simplicité des problèmes à résoudre. En effet, chaque deux classes prises à part sont moins chevauchées que toutes les classes.

II.5.2. Une-contre-reste (1vsR)

Selon la formulation de Vapnik [9], elle consiste à déterminer pour chaque classe k un hyperplan $H_k(x)$ la séparant de toutes les autres classes. Cette classe k est considéré comme étant la classe positive (+1) et les autres classes comme étant la classe négative (-1), ce qui résulte, pour un problème à K classes. Un hyperplan H_k est défini pour chaque classe k par la fonction de décision suivante :

$$k^* = \arg_{1 \leq k \leq K} \text{Max}(f_k(x)) \quad (\text{II.23})$$

avec $f_k(x)$: la fonction de décision (k classe) de la sortie MLP ou SVM.

La Figure II.15 représente un exemple d'un cas de séparation de 3 classes.

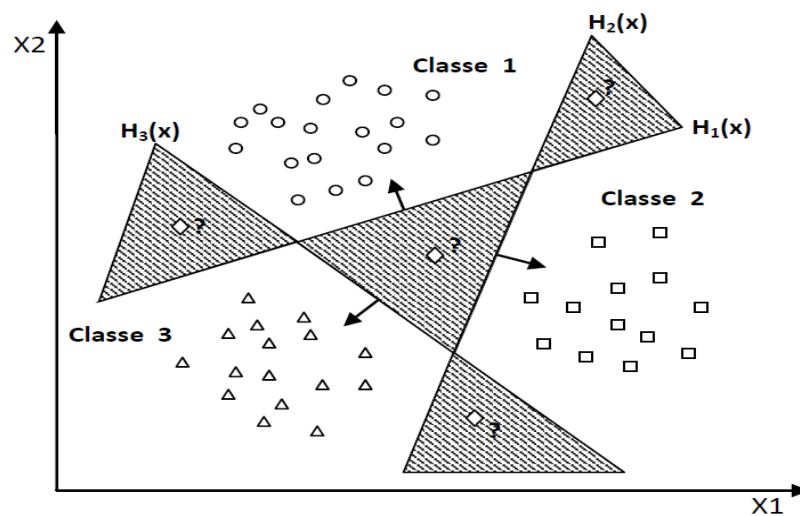


Figure II. 15. Séparation entre trois classes en utilisant la méthode 1vsR.

II.5.3. Graphe de décision

C'est une méthode développée par Platt et al [49] pour résoudre le problème des zones d'indécision dans la méthode 1vs1. Premièrement, l'entraînement est effectué par la même méthode 1vs1 de la section précédente pour obtenir $K(K-1)/2$ hyperplans. Puis, au lieu d'utiliser le vote pour l'affectation des nouveaux exemples, on construit un graphe de décision. Pour cela, on définit une mesure E_{ks} de la capacité de généralisation sur les différents hyperplans obtenus c.-à-d pour chaque paire de classes. Cette mesure représente le rapport entre le nombre de vecteurs supports de l'hyperplan et le nombre d'exemples des deux classes correspondantes (équation II.24) [47].

$$E_{ks} = \frac{N_{ks}}{N} \quad (\text{II.24})$$

Ou N_{ks} est un nœud de graphe étiqueté de k et s classe, N est le nombre d'exemples. On obtient ainsi un graphe de décision similaire à l'exemple de la Figure II.16 dont les feuilles sont les classes et les noeuds internes sont les hyperplans:

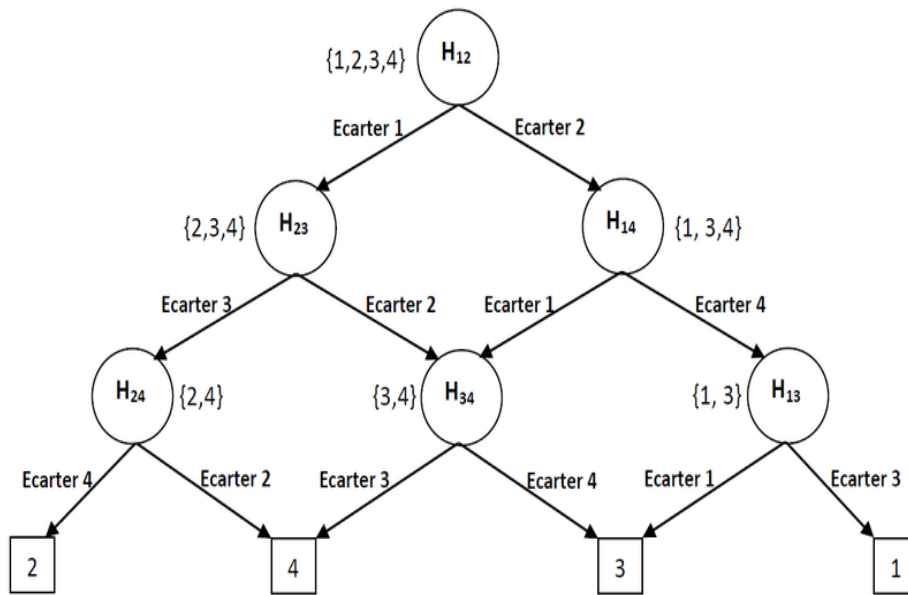


Figure II. 16. Graphe de décision acyclique orienté à quatre classes.

II.5.4. SVM basées arbres de décision

Dans cette méthode, on apprend pour K classes, $(K-1)$ hyperplans. Chaque hyperplan sépare une ou plusieurs classes du reste, selon un découpage choisi. On peut choisir, par exemple un découpage semblable à la méthode 1vsR où l'hyperplan H_i sépare la classe i des classes $i + 1, i + 2, \dots, K$ (voir la Figure II.17).

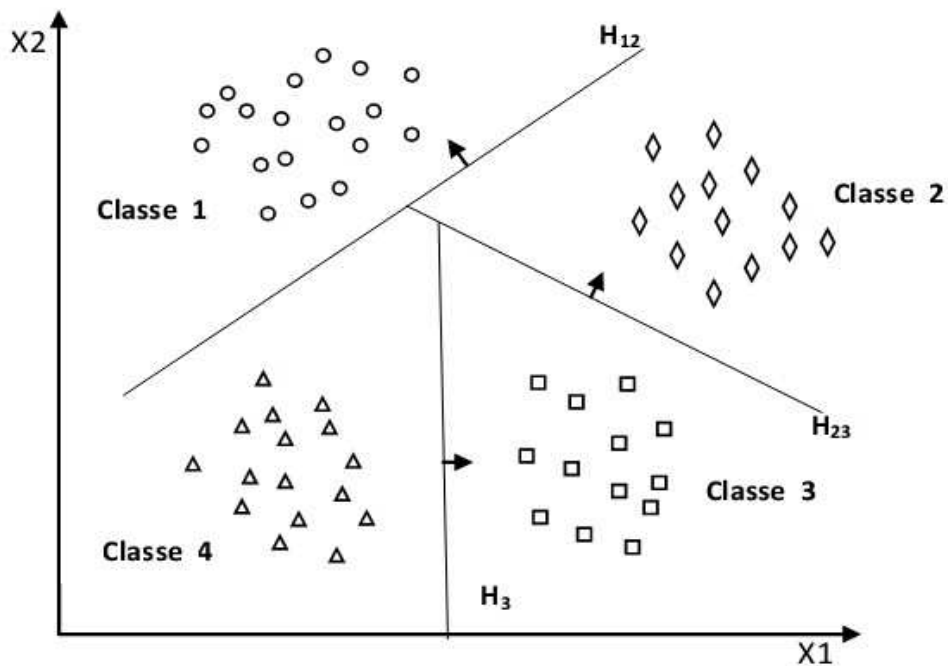


Figure II. 17. SVM multiclassé par arbre de décision.

II.6. Conclusion

Dans ce chapitre, nous avons présenté le contexte de notre travail où nous avons vu un aperçu général sur les réseaux de neurones et la méthode de machines à vecteurs de support et détailler d'avantage les notions auxquelles nous avons fait appel pour élaborer notre travail. Et présenté les propriétés fondamentales des réseaux de neurones qui justifient l'intérêt croissant qui leur est accordé et que sont capable d'intervenir dans la résolution de nombreux problèmes de la classification.

Nous sommes intéressés par la méthode de la localisation avec l'utilisation des méthodes de l'intelligence artificielle (MLP, RBFN, SVM) sont examinés dans le chapitre III.

Chapitre III

Résultats de simulations

III.1. Introduction

La localisation de l'environnement intérieur fait aujourd'hui partie de notre quotidien. Elle permet de guider des usagers, trouver rapidement certains services de proximité, assurer la surveillance, etc. Plusieurs solutions existent d'ores et déjà en extérieur, basées pour la majorité sur le système GPS. À l'intérieur d'un bâtiment (localisation dite indoor), la forte atténuation des signaux GPS, provenant directement des satellites, favorise le déploiement d'une infrastructure locale. Parmi les technologies couramment retenues, WiFi offre plusieurs avantages. Bon marché, de faible complexité et consomme peu d'énergie, WiFi est également répandu sur une vaste gamme d'équipements.

Comme la propagation des ondes radiofréquence dans un environnement intérieur n'est pas aisément modélisable, on ne peut pas déduire de données de localisation ou distance précise à partir des RSSI (voir Chapitre I). La méthode la plus courante, est de créer une base de données de vecteurs RSSI correspondant à une discrétisation des positions dans un environnement fermé (exemple : bâtiment). Le principe de la technique de FingerPrinting est de trouver la meilleure correspondance entre le vecteur RSSI mesuré par l'utilisateur et ceux conservés dans une base de données, et d'estimer la position à partir de cette correspondance.

Dans ce cadre, nous avons utilisé des modèles d'intelligence artificielle (réseaux de neurones, et machine à vecteurs de support) pour l'objectif de l'estimation ou la prédiction de la localisation d'un utilisateur dans un environnement fermé.

Dans ce chapitre, nous nous sommes concentrés sur les étapes de la description de la base de données, la localisation avec l'utilisation des techniques de l'intelligence artificielle, et la dernière étape est la comparaison entre les différentes méthodes de classification.

III.2. Description de la base de données

Pour prédire l'emplacement de l'utilisateur avec précision. Un modèle défini et cohérent à former et à déployer dans un dispositif de suivi ou de surveillance. Nous mesurons la l'indicateur de puissance du signal WIFI reçu de divers routeurs dans une localisation délimitée et ensuite entraînons des modèles d'intelligence artificielle (réseaux de neurones, et machine à vecteurs de support) afin qu'il puisse davantage de prédire l'emplacement de l'utilisateur pour un ensemble de données inconnu ayant des intensités de signal (RSSI).

Dans ce cadre notre travail a pour objectif de mettre en œuvre un système de positionnement à l'intérieur d'un espace fermé, où l'utilisateur est menu d'une ESP32 ou une carte Raspberry Pi (voir Chapitre IV), pour l'objectif de calculer l'indicateur RSSI. Cela est effectué en basant

sur les méthodes de l'intelligence artificielle. Pour tester les performances de notre méthode proposée, nous avons utilisé la base de données qui est présentée dans l'article [50].

Nous considérons une base de données d'un environnement fermé dispose de 7 routeurs WiFi et ses puissances de signal reçues de ces routeurs catégorisent l'emplacement de l'utilisateur dans la salle de conférence, la cuisine ou la salle de sport intérieure ou le reste de l'environnement. Les classes des emplacements correspondantes respectivement étiquetées 1, 2, 3 et 4.

Le Tableau III.1 illustre un exemple de la base de données. $RSSI_1$ correspond à l'indicateur de puissance du signal reçu du routeur 1, $RSSI_2$ correspond à l'indicateur de puissance du signal reçu du routeur 2, et de même pour les autres routeurs. Ces mesures sont formulées dans un ensemble de données pour la classification en considérant les sept routeurs sans fil comme les dimensions d'entrée [$RSSI_1$ $RSSI_2$,..., $RSSI_7$], qui sont utilisées pour prédire l'emplacement de l'utilisateur dans un environnement fermé. Dans cette base de données 2000 observations est effectuée dans l'environnement fermé pour sept routeurs différents.

La Figure III.1 représente un exemple de la division de la base de données sur deux ensembles : apprentissage et test, respectivement. Le nombre total des échantillons dans les deux ensembles (apprentissage, et test) sont respectivement 1333 et 667. Après avoir préparé un ensemble de données concret, nous entraînons notre modèle avec l'utilisation de trois méthodes différentes (MLP, RBFN et SVM) qui sera détaillé dans la section qui suit.

La Figure III.2 représente des exemples de la base de données de RSSI en deux dimensions en fonction de leurs emplacements (classe). D'après cette figure, on trouve un problème de discrimination entre les différentes classes. Pour cela nous avons fait plusieurs expériences pour choisir le meilleur modèle.

Tableau III. 1. Exemple de la base de données pour la localisation.

$RSSI_1$	$RSSI_2$	$RSSI_3$	$RSSI_4$	$RSSI_5$	$RSSI_6$	$RSSI_7$	Classe
-63	-60	-60	-67	-76	-85	-84	1
-61	-60	-68	-62	-77	-90	-80	1
-42	-53	-62	-38	-66	-65	-69	2
-44	-55	-61	-41	-66	-72	-68	2
-48	-54	-50	-49	-61	-81	-84	3
-54	-53	-54	-50	-63	-79	-77	3
-61	-52	-49	-56	-46	-84	-83	4
-55	-50	-51	-61	-48	-82	-79	4

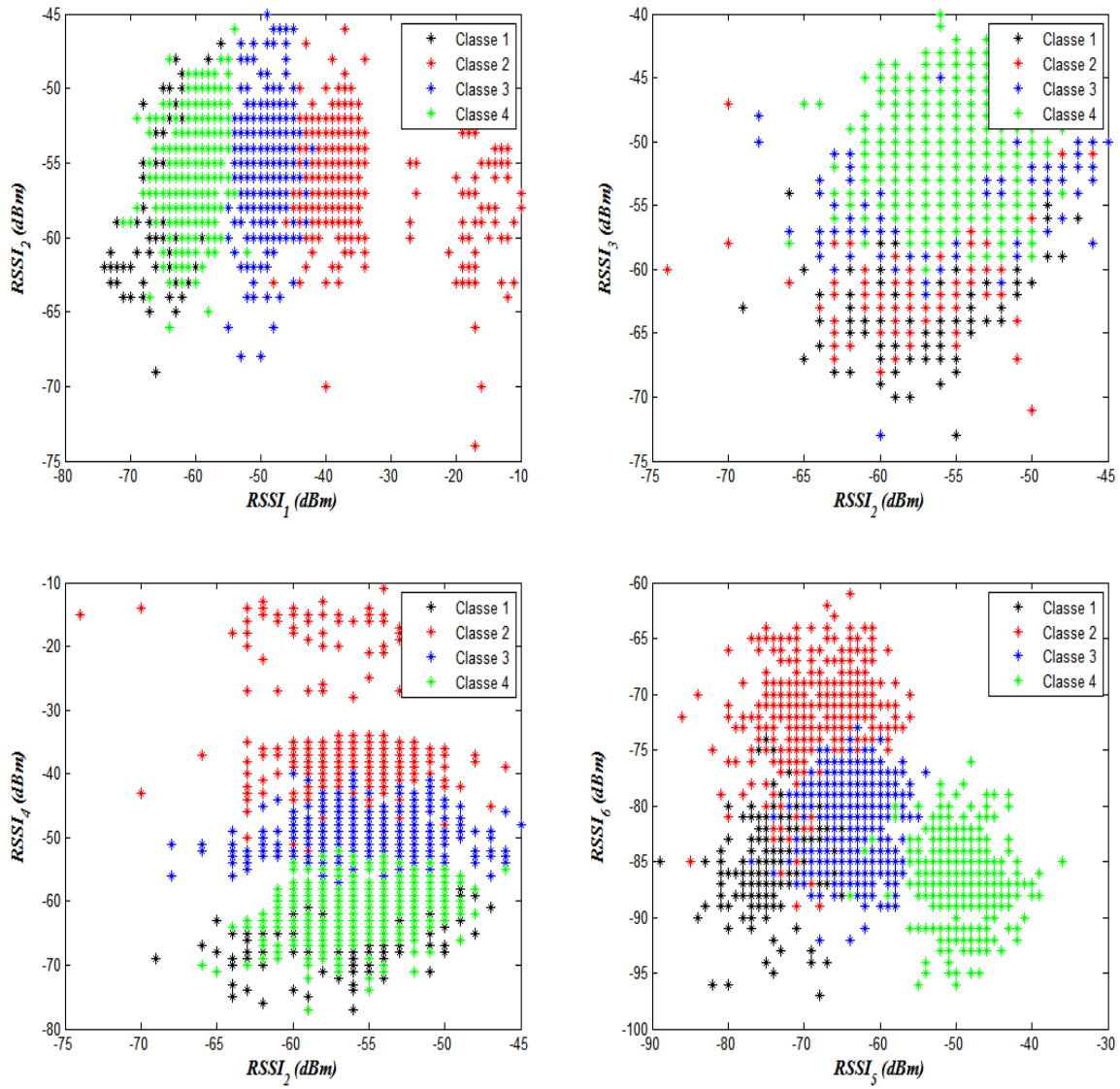


Figure III. 1. Exemples de la base de données.

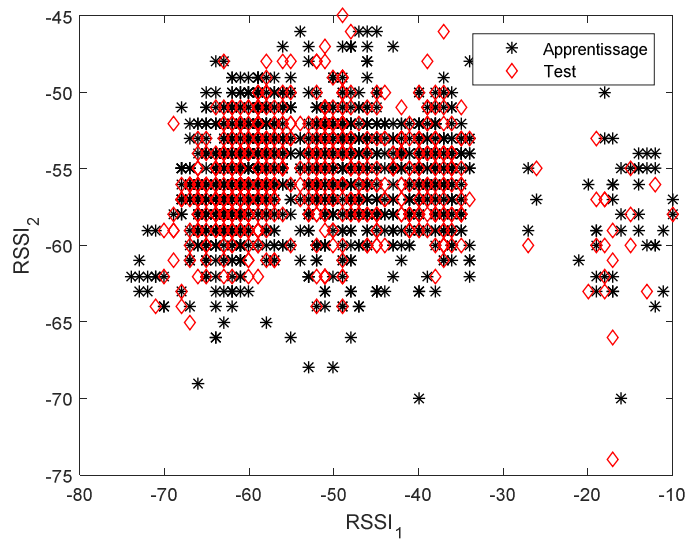


Figure III. 2. Exemple de la division de la base de données en deux ensembles : Apprentissage et test.

III.3. Méthode proposée

Notre méthode proposée consiste à effectuer des mesures puis construire un modèle de classification pour l'objectif de la localisation d'un utilisateur dans un environnement intérieur. Notre approche comprend donc deux étapes: une étape de calibrage ou d'apprentissage et l'étape d'estimation de la position (localisation indoor). Une des méthodes les plus efficaces est la méthode d'empreinte radio (en anglais RF FingerPrinting), qui se base sur les mesures des puissances reçues (RSSI Received Signal Strength Indicator) par l'utilisateur à partir des points d'accès WiFi. La technique d'empreinte radio, se fait en deux phases :

Phase de calibrage: Cette phase consiste à relever les caractéristiques (FingerPrint) des signaux aux différents endroits d'un environnement intérieur (exemple : bâtiment), et les enregistrer dans une base de données pour pouvoir être exploitées par la suite dans une phase de localisation ou classification en utilisant les réseaux de neurones (MLP, RBFN) ou la méthode basée sur la machine à vecteur de support (voir la Figure III.3).

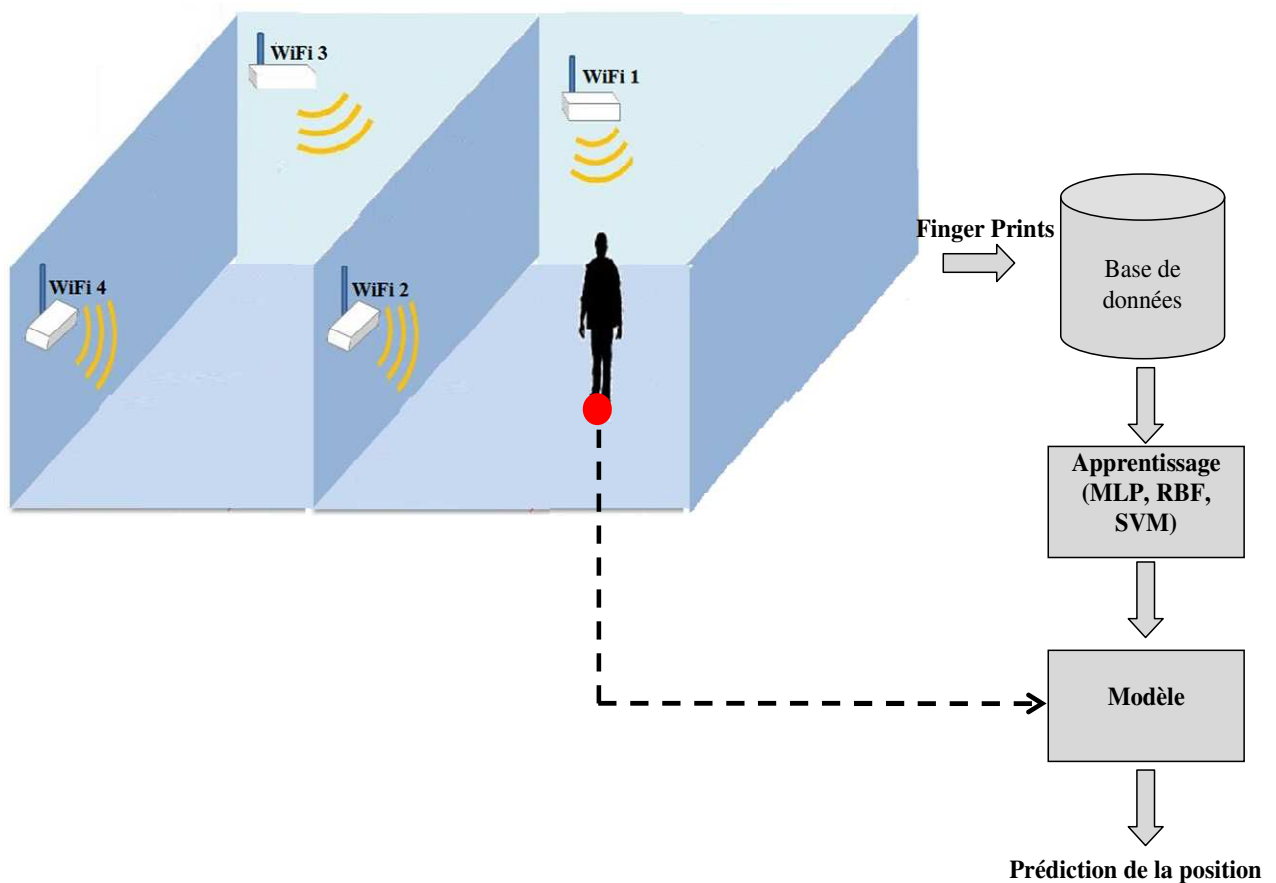


Figure III. 3. Localisation par FingerPrint dans un réseau WiFi.

Pour construire un fingerprint du signal WiFi, il faut utiliser une unité de mesure à la fois applicable à tous les modèles de routeurs et facilement mesurable sur un ordinateur, un smartphone, ou des circuits contenant la technologie WiFi (voir Chapitre IV). C'est pour ces raisons que généralement, la puissance d'un signal WiFi se mesure en *dBm* (décibels relatifs par milliwatt) :

- L'échelle est comprise entre -30 et -90 *dBm*.
- À -30 *dBm*, la puissance est excellente.
- Entre -50 et -60 *dBm*, la qualité du signal WiFi est bonne.
- À -90 *dBm*, la qualité du signal WiFi est mauvaise.
- Les mesures RSSI représentent la qualité relative d'un signal reçu sur un dispositif. RSSI indique le niveau de puissance reçu après une possible perte au niveau de l'antenne et du câble. Plus la valeur RSSI est élevée, plus le signal est fort. Lorsque mesuré avec des chiffres négatifs, si le chiffre est proche de zéro ceci signifie qu'il existe un meilleur signal. Par exemple : -50 est un assez bon signal, -75 est un signal plutôt raisonnable et -100 n'est pas un signal du tout.

L'indicateur de puissance RSSI et path-loss est donné par l'expression suivante:

$$RSSI = -10 \times n \times \log_{10}(d) + c \quad (III.1)$$

où l'exposant de perte de trajet n et la constante c étaient requis pour chacune des technologies de communication sans fil dans chacun des scénarios de test pour déterminer les valeurs correspondantes.

Il est clair que la fiabilité d'un système de localisation repose en grand porteur sur le choix de la méthode de classification. L'étude que nous avons parité dans les sous sections suivantes, nous a permis de faire le choix des combinaisons WiFi-MLP, WiFi-RBFN et WiFi-SVM pour la classification et la positionnement de l'utilisateur dans un environnement fermé.

Phase de positionnement: Appelée aussi la localisation en temps réel, consiste à déterminer la position de l'objet ou l'utilisateur en utilisant le modèle d'intelligence artificielle trouvé dans la phase de calibrage; plusieurs mesures successives de puissance doivent être réalisées pour une meilleure évaluation de la position, les ondes radios étant affectées par plusieurs facteurs.

III.4. Localisation par réseaux de neurones artificiels et machine à vecteurs de support

Dans cette partie nous avons utilisé trois types de classificateurs (MLP, RBFN et SVM) pour faire la localisation d'un utilisateur dans un environnement indoor. Un système de

classification opère en deux phases consécutives: une phase de conception et une phase de généralisation. La première phase consiste à choisir l'architecture optimale du modèle pour pouvoir s'adapter à différentes conditions d'entrée, l'application de cette phase est en mode hors ligne. Nous avons opté pour la normalisation des signaux RSSI. Une fois l'apprentissage du modèle est terminé, il passe en phase de généralisation pour réaliser la localisation en temps réel.

La normalisation est un prétraitement effectué sur tous les signaux RSSI. Elle consiste à normaliser l'amplitude du signal entre -1 et 1. La normalisation est calculée selon l'équation suivante :

$$RSSI_N = \frac{RSSI - \min(RSSI)}{\max(RSSI) - \min(RSSI)} \times 2 - 1 \quad (III.2)$$

avec $RSSI_N$ est le signal RSSI normalisée.

III.4.1. Évaluation des performances

III.4.1.1. Taux de classification

Dans le domaine de la classification, nous avons besoin d'un critère de précision. Dans notre cas nous avons choisi de calculer le taux de classification, il est donné par la formule suivante:

$$Acc(\%) = \frac{P}{N} \times 100 \quad (III.3)$$

P : Echantillons bien classées.

N : Nombre d'échantillons des classes cibles.

Dans le cas de la technique *k-fold-cross-validation*, le taux de classification sur l'ensemble de k validation, est donné par l'équation suivante:

$$Acc(\%) = \frac{1}{k} \sum_{k=1}^k \frac{P_k}{N_k} \times 100 \quad (III.4)$$

Ou $Acc_k(\%)$ est le taux de classification sur k^{eme} validation, P_k est les échantillons bien classés et N_k leur nombre d'échantillons sur l'ensemble k validation.

III.4.1.2. Matrice de confusion

La matrice de confusion, dans la terminologie de l'apprentissage supervisé, est un outil servant à mesurer la qualité d'un système de classification. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence).

Un des intérêts de la matrice de confusion est qu'elle montre rapidement si le système parvient à classifier correctement. Un modèle sans erreurs aura ses résultats concentrés sur la diagonale de sa matrice de confusion.

III.4.2. Classification par réseaux de neurones artificiels

Notre première méthode est la classification de réseaux de neurones multicouche MLP par l'utilisation des données de l'indicateur du signal WiFi. L'estimation de la meilleure architecture de l'MLP est obtenue par la technique de *k-fold-cross-validation* avec *k* est égale à 5.

La variation des paramètres du model MLP est donnée par le tableau suivant:

Tableau III. 2. Architecture du modèle MLP.

	Minimum	Maximum
Nombre de neurone dans le couche cachée 1 (NN1)	10	100
Nombre de neurone dans le couche cachée 2 (NN2)	10	100
Taux d'apprentissage (μ)	0.1	0.9
Momentum (m)	0.1	0.9

Pour chaque combinaison des paramètres de classifieur MLP, les précisions (*Acc*) sur la base de validation sont sauvegardées en fonction de NN1, NN2, μ et m , comme montres les figures III.4 à III.6.

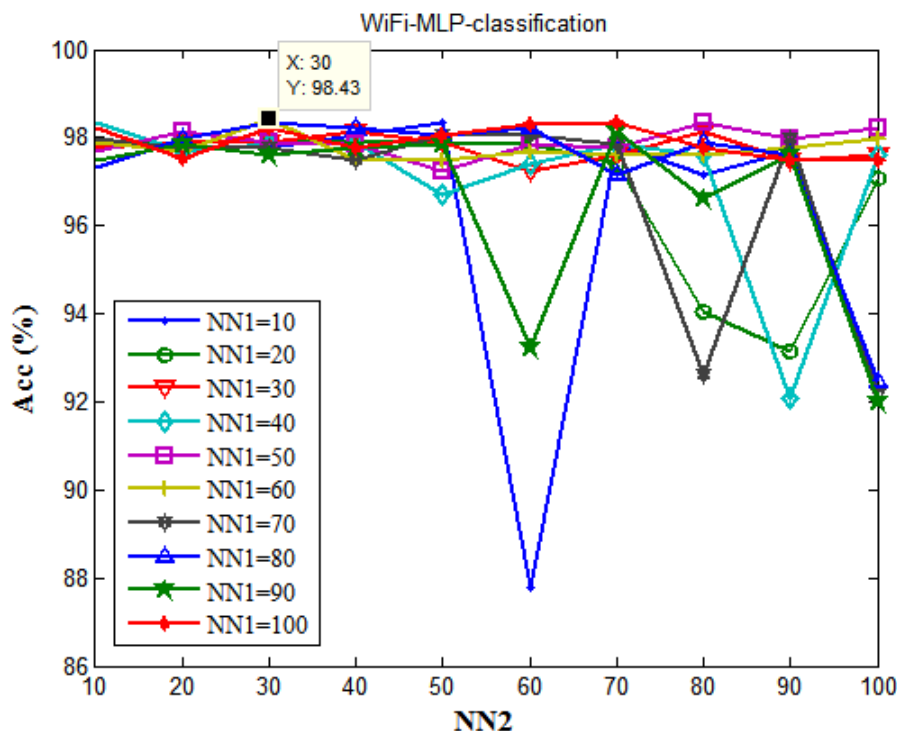


Figure III. 4. Précision de classifieur MLP sur l'ensemble de validation (*Acc*) en fonction des nombres de neurones dans les couches cachées.

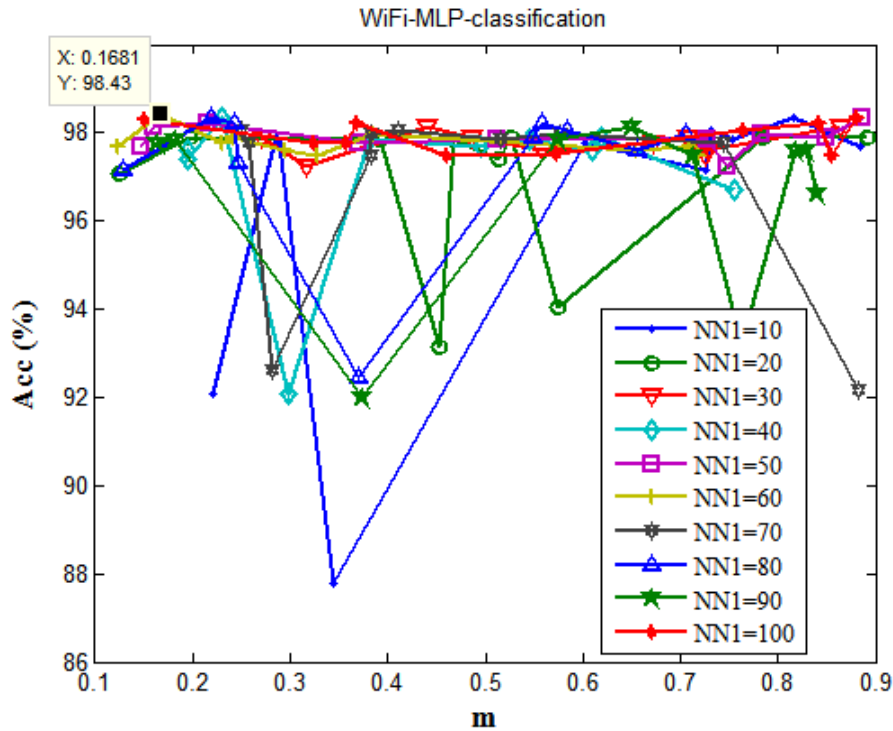


Figure III. 5. Précision de classifieur MLP sur l'ensemble de validation (Acc) en fonction des nombres de neurones dans la première couche cachée et le paramètre d'accélération de réseaux de neurone momentum.

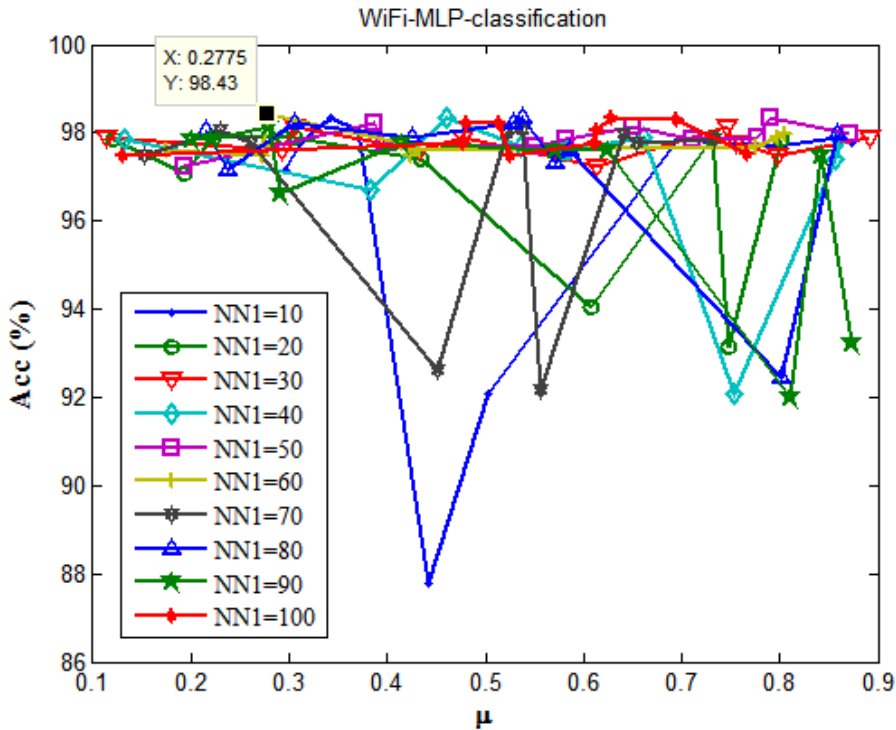


Figure III. 6. Précision de classifieur MLP sur l'ensemble de validation (Acc) en fonction des nombres de neurones dans la première couche cachée et le taux d'apprentissage.

Dans les figures III.4, III.5 et III.6, nous avons représentés toutes les valeurs de précision. Le meilleur modèle de classifieur MLP est obtenu pour les paramètres : le nombre de neurones dans la première couche cachée NN1 est 60, dans la deuxième couche cachée NN2=30, le taux d'apprentissage est 0.1681 et le momentum 0.2775, le réseau donne une précision maximale égale à 98.43% sur l'ensemble de validation.

Une fois la meilleure architecture sélectionnée, nous utilisons l'ensemble de test pour tester les performances du classifieur MLP. La Figure III.7 représente la matrice de confusion pour l'ensemble de test, la précision sur le test est 97.27%.

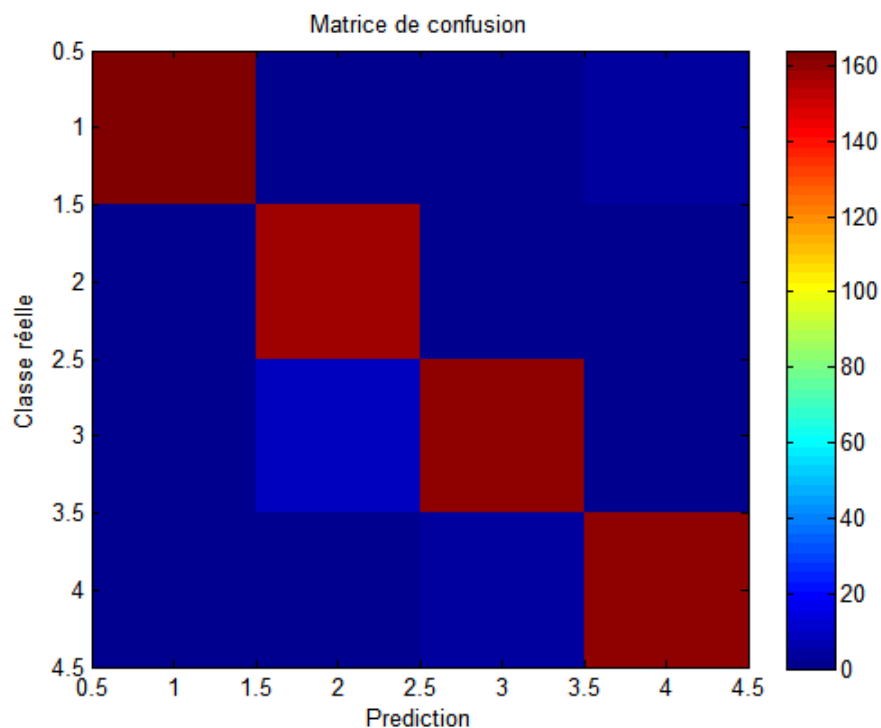


Figure III. 7. Matrice de confusion sur l'ensemble de test (MLP).

La meilleure architecture de l'RBFN est obtenue par la technique *k-fold-cross-validation*. En faisant varier le largeur de la gaussienne σ entre 0.1 à 1 avec un pas de 0.1, et le nombre de neurones dans la couche cachée NN entre 10 à 100 avec un pas de 10.

La Figure III.8 représente la variation de la précision (Acc) sur l'ensemble de validation en fonction de nombre de neurones dans la couche cachée et σ . Le réseau RBFN donne une précision maximale égale à 98.21% avec l'architecture : nombre de neurones dans la couche cachée NN=100, et la largeur de gaussienne égale à 0.7.

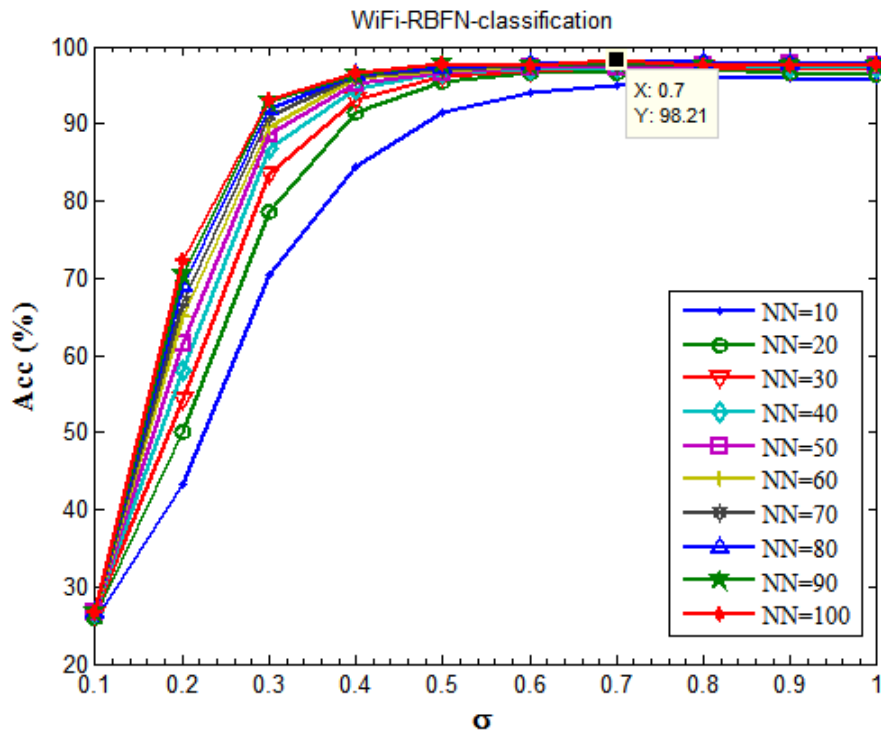


Figure III. 8. Précision (*Acc*) de réseaux de neurones RBFN en fonction de nombre de neurones dans la couche cachée et la largeur de gaussienne σ .

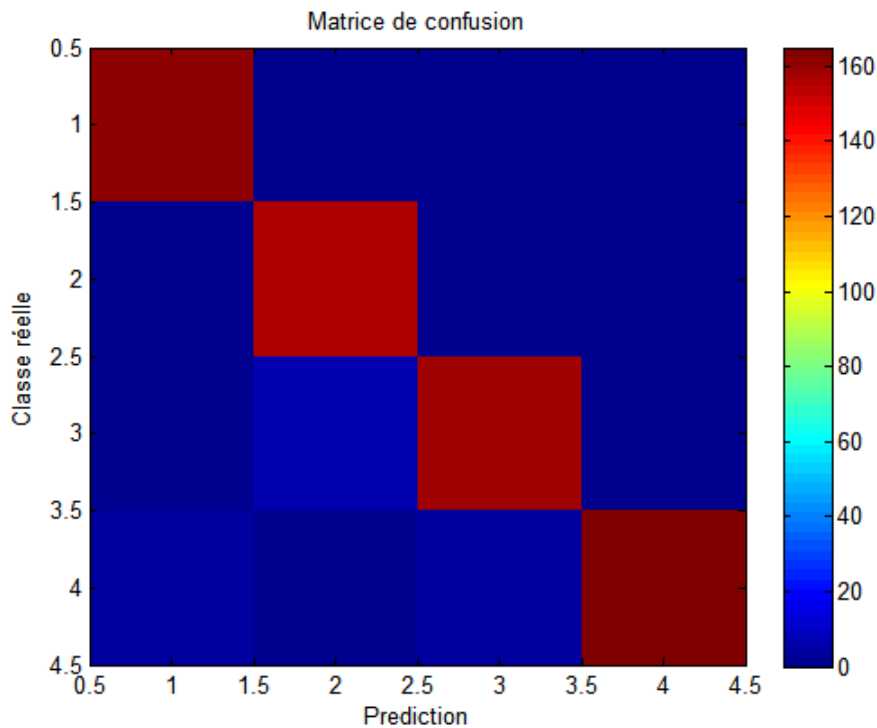


Figure III. 9. Matrice de confusion de la meilleure architecture RBFN sur l'ensemble de test. Une fois le meilleur modèle est obtenu, nous utilisons l'ensemble de test pour tester les performances du classificateur RBFN. La Figure III.9 représente la matrice de confusion sur l'ensemble de test. La précision sur l'ensemble de test est 97.12%.

III.4.3. Classification par machine à vecteurs de support

La troisième méthode est l'apprentissage de la machines à vecteurs de support est obtenu par la technique de validation *k-fold-cross-validation* avec *k* est égale à 5. La variation des paramètres du model SVM est donnée par le tableau III.3.

Tableau III. 3. Variation des paramètres à optimisés par le modèle SVM.

	Minimum	Maximum
La régularisation C	0.001	1000
Largeur de la gaussienne γ	0.001	1

La Figure III.10 représente le taux de classification en fonction des paramètres à optimiser: C et γ , avec l'utilisation de base de données WiFi.

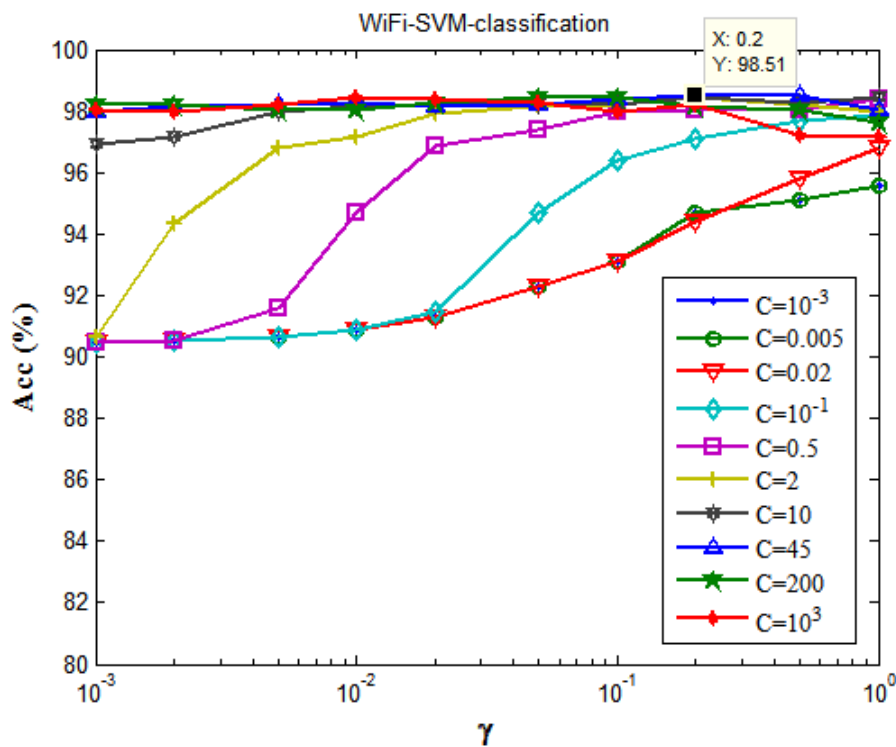


Figure III. 10. Précision (Acc) en fonction de C et γ .

La Figure III.10 illustre les combinaisons des paires (C ; γ) qui donne les meilleures précisions sur l'ensemble de validation. Le meilleur taux de classification ($Acc=98.51\%$) est obtenu avec les combinaisons : $C=45$, et $\gamma=0.2$.

Une fois le meilleur modèle est obtenu, nous utilisons l'ensemble de test pour tester les performances du classificateur SVM. La Figure III.11 représente la matrice de confusion sur l'ensemble de test. La précision sur l'ensemble de test est 98.18%.

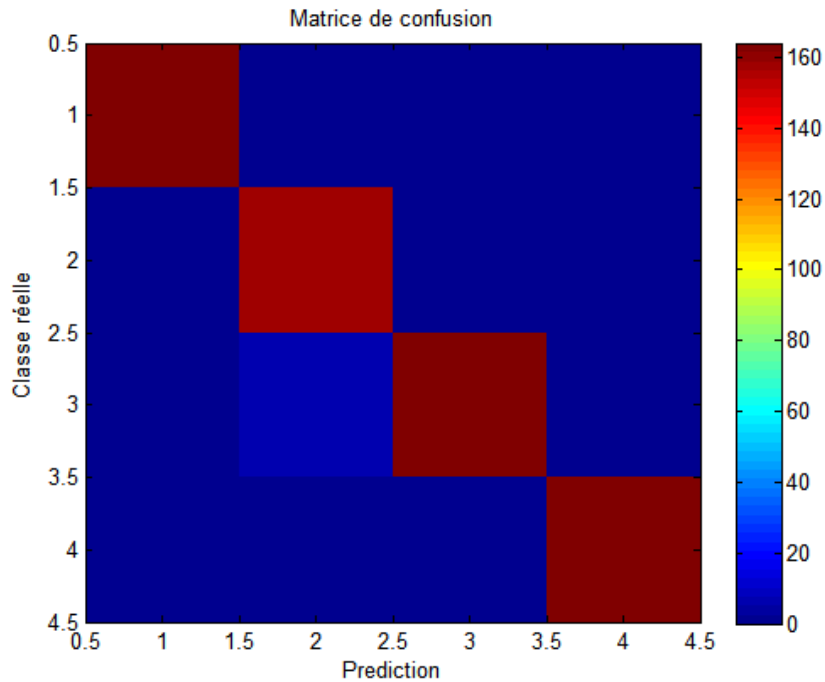


Figure III. 11. Matrice de confusion sur l'ensemble de test (SVM).

III.4.4. Etude comparative

Dans cette section nous allons exposer les résultats comparatifs obtenus par l'application des trois méthodes proposées en utilisant en premier lieu les réseaux de neurones artificielles (MLP, RBFN) et en deuxième lieu machine à vecteurs de support SVM.

La figure suivante présente une comparaison entre les meilleurs résultats obtenus par les trois méthodes de classification (MLP, RBFN et SVM) sur les ensembles de validation et de test. Nous remarquons que le classificateur SVM donne des résultats significativement meilleurs que les méthodes MLP et RBFN.

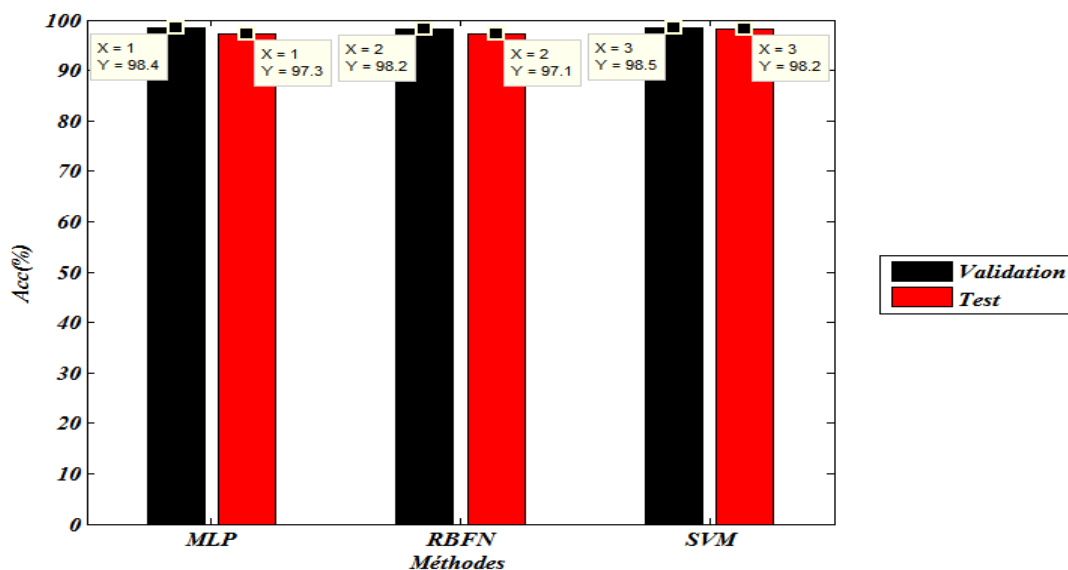


Figure III. 12. Etude comparative entre MLP, RBFN et SVM.

III.5. Conclusion

Dans ce chapitre, nous avons expliqué les deux phases essentielles pour la localisation des utilisateurs dans un environnement intérieur (indoor); plus précisément : la phase de calibrage et la phase de localisation en temps réel. D'après les résultats obtenus, l'algorithme basé sur le classifieur machine à vecteurs de support (SVM) semble être le plus efficace en terme de taux de classification sur les deux ensembles de validation et de test.

Dans le chapitre suivant, nous allons présenter la partie pratique de notre travail avec l'utilisation de deux cartes de traitement telles que Arduino, Raspberry *Pi* et une autre carte équipée du WiFi (ESP32) pour l'objectif de la création des bases de données pour la localisation.

Chapitre IV

Réalisation pratique

IV. 1. Introduction

Dans les chapitres précédents on a travaillé sur le côté théorique du projet, en revanche dans ce dernier chapitre on travaille sur deux côté logiciel et matériel. Nous avons présenté les différentes étapes qui nous ont permis de réaliser un système de localisation indoor en utilisant la technologie WiFi. Dans ce cadre de notre travail, nous avons proposés trois montages pour résoudre le problème de la localisation d'un utilisateur dans un environnement intérieur, le premier montage à base de la localisation GPS, et deux montages basés sur la technologie WiFi (ESP32, Raspberry Pi).

IV. 2. Système de localisation GPS

Pour réaliser notre système de localisation à base de GPS, nous avons besoin une carte Arduino et un module GPS. Dans les sous sections suivantes, nous détaillons les composants nécessaire pour notre montage (la carte Arduino UNO, et le module GPS). Le brochage de notre montage de localisation est fait en utilisant le logiciel de dessin des schémas électroniques “*fritzing*”.

IV.2.1. Carte Arduino

Le système Arduino donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, pour programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le cout de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique [51].

Ce système est constitué de deux parties (matériel et logiciel):

IV.2.1.1. Partie Hardware (Le matériel)

Les cartes électroniques dont les schémas sont en libre circulation sur internet. Dans la section suivante, nous détaillons la carte Arduino UNO.

La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. Sa mémoire de 32KB et son grand nombre d'E/S font de ce circuit compatible. Ce type de carte est un élément idéal pour les systèmes embarqués ou pour notre projet qui nécessite des applications multitâches. L'Arduino Uno peut se programmer avec le logiciel IDE Arduino.

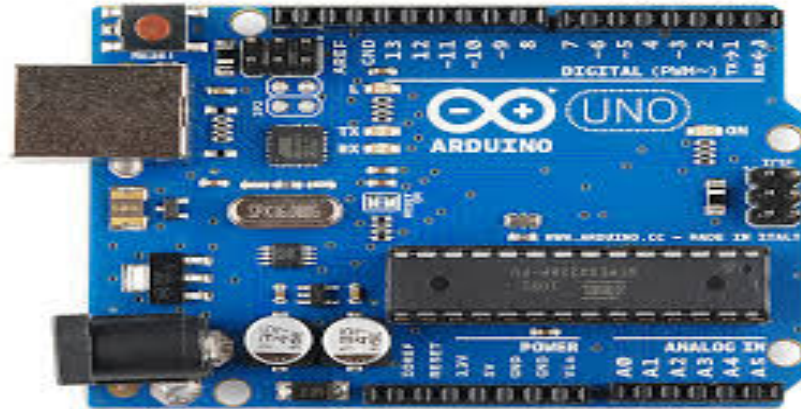


Figure IV. 1. Carte Arduino Uno.

Les spécifications techniques de cette carte Arduino Uno sont les suivantes :

- ❖ Microcontrôleur Atmel ATmega328.
- ❖ Voltage opérationnel (au niveau logique) : 5V.
- ❖ Pins d'entrées/sorties digitales : 14 (dont 6 disposent d'une sortie PWM).
- ❖ Pins d'entrée analogique : 6 (utilisables en broches E/S numériques).
- ❖ Courant direct par pin d'entrée/sortie : 40mA.
- ❖ Mémoire Flash : 32KO (ATmega328) dont 2KO sont utilisés par le bootloader.
- ❖ SRAM : 2KO (ATmega328).
- ❖ EEPROM : 1KO (ATmega328).
- ❖ Vitesse d'horloge : 16MHz.

La carte Arduino UNO peut être alimentée par l'USB ou par une alimentation externe. La source est sélectionnée automatiquement.

La tension d'alimentation extérieure (hors USB) peut venir soit d'un adaptateur AC-DC ou de piles. L'adaptateur peut être connecté grâce à un 'jack' de 2.1mm positif au centre. Le raccordement vers un bloc de piles peut utiliser les bornes GND et *Vin* du connecteur d'alimentation (POWER). La carte peut fonctionner à l'aide d'une tension extérieure de 7 à 12volts.

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino (IDE) peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

IV.2.1.2. Partie logiciel (Software)

Le logiciel IDE Arduino est gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.

Pour commander un Arduino sur PC on installe le logiciel IDE Arduino (par exemple la version 1.6.5 dans notre cas) qui est une application Java et qui a l'interface montrée dans la figure IV.2. IDE Arduino est une interface souple et simple est exécutable sur n'importe quel système d'exploitation. Arduino basé sur la programmation en C ou C++ [51].

Le logiciel Arduino a pour les fonctions principales :

- De pouvoir écrire et compiler des programmes pour la carte Arduino.
- De se connecter avec la carte Arduino pour y transférer les programmes.
- De communiquer avec la carte Arduino.

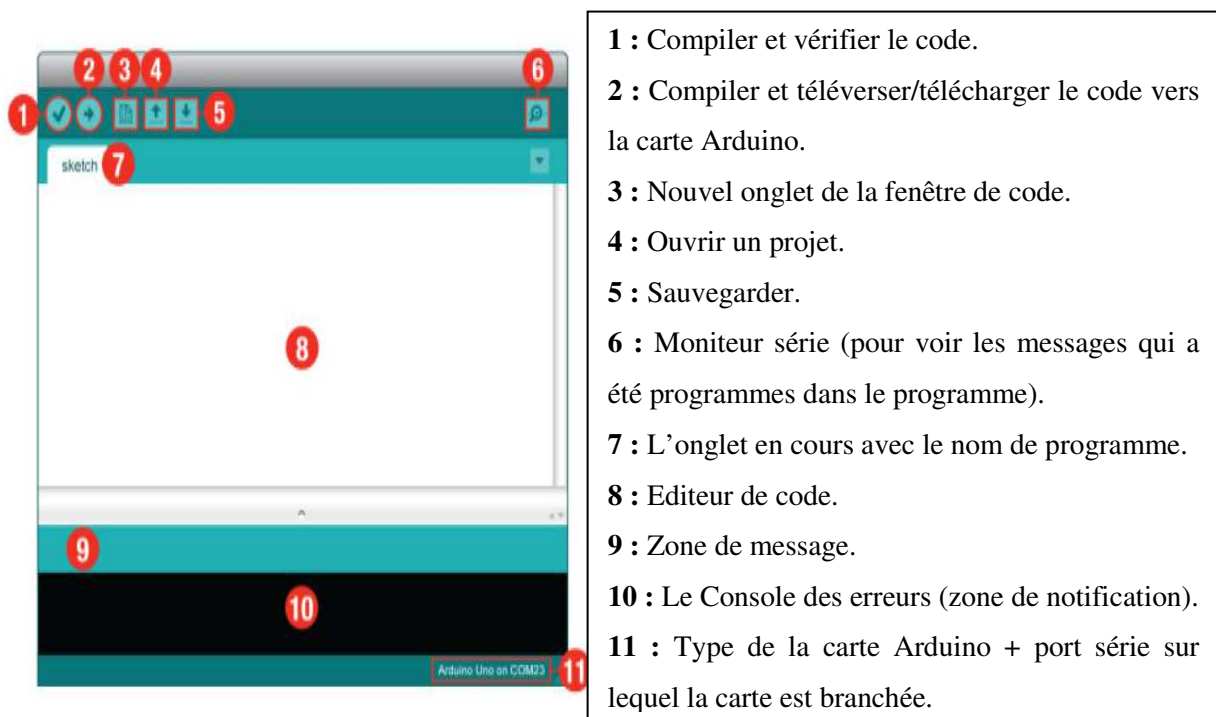


Figure IV. 2. Description de l'Interface Arduino IDE.

La structure d'un programme IDE Arduino est comporte trois parties [51]:

- (1) La partie déclaration des variables (optionnelle).
- (2) La partie initialisation et configuration des entrées/sorties : la fonction **setup ()**.
- (3) La partie principale qui s'exécute en boucle : la fonction **loop ()**.

```

Fichier Édition Croquis Outils Aide
DigitalReadSerial $
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);    // delay in between reads for stability
}
11 Arduino/Genuino Uno on COM14

```

Figure IV. 3. Structure d'un programme Arduino.

IV.2.2. Modules GPS

Global Positioning System (GPS) est un système de navigation par satellite composé d'au moins 24 satellites. Le GPS fonctionne dans toutes les conditions météorologiques, partout dans le monde, 24h/24h.

Il existe plusieurs Shields Arduino GPS mais dans notre projet nous avons utilisé le module GPS avec la référence GY-NEO6MV3 avec une antenne externe, donc nous devons le souder (voir figure IV.4). Ce module prend en charge le débit de 4800 bps à 230400 bps avec un baud par défaut de 9600 est utilisé pour recevoir les données des satellites. Les données sont envoyées à l'Arduino qui lit les coordonnées en extrayant la chaîne \$GPGGA (Global Positioning system fix data).

Il y a une LED sur le module GPS GY-NEO6MV3 qui indique l'état de la position fixe. Il clignotera à différents taux selon l'état dans lequel il se trouve (voir la figure IV.13).

- ✓ Pas de clignotement : signifie qu'il recherche des satellites.
- ✓ Clignote toutes les 1 s : signifie que la correction de position est trouvée.

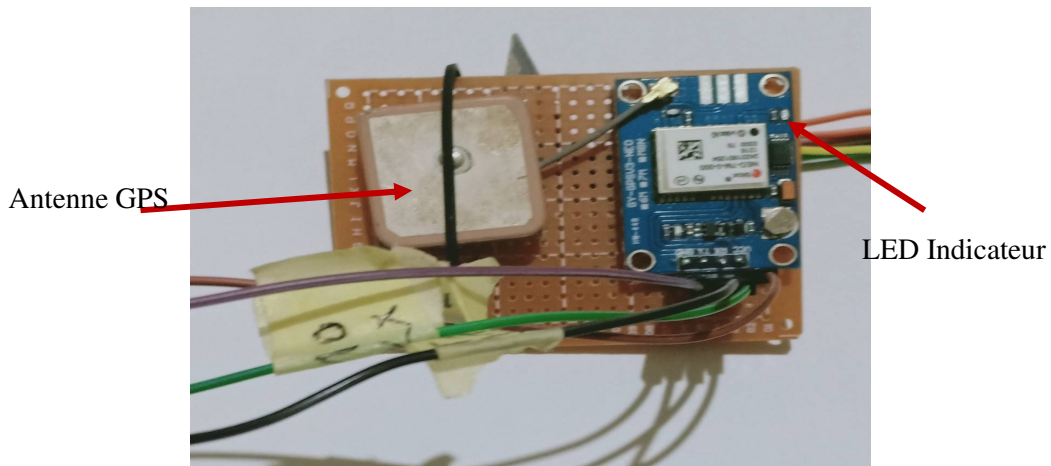


Figure IV. 4. Module GPS (GY-NEO6MV3).

IV.2.2.1. Branchement du module GPS

Le GPS utilise 4 pins, VCC et GND et ensuite TX et RX qui sont connectés à un Arduino. Ces deux derniers pins sont respectivement le signal de transmission et celui de réception. Les Figures IV.5 et IV.6 représentent le branchement et le circuit électrique de module GPS avec l'arduino Uno.

Tableau IV. 1. Affectation des broches GPS.

Nom de broche	Rôle de la broche
GND	La broche de terre (0V).
TX (émetteur)	Utilisé pour la communication série.
RX (récepteur)	Utilisé pour la communication série.
VCC	Alimente le module GPS (+5V).

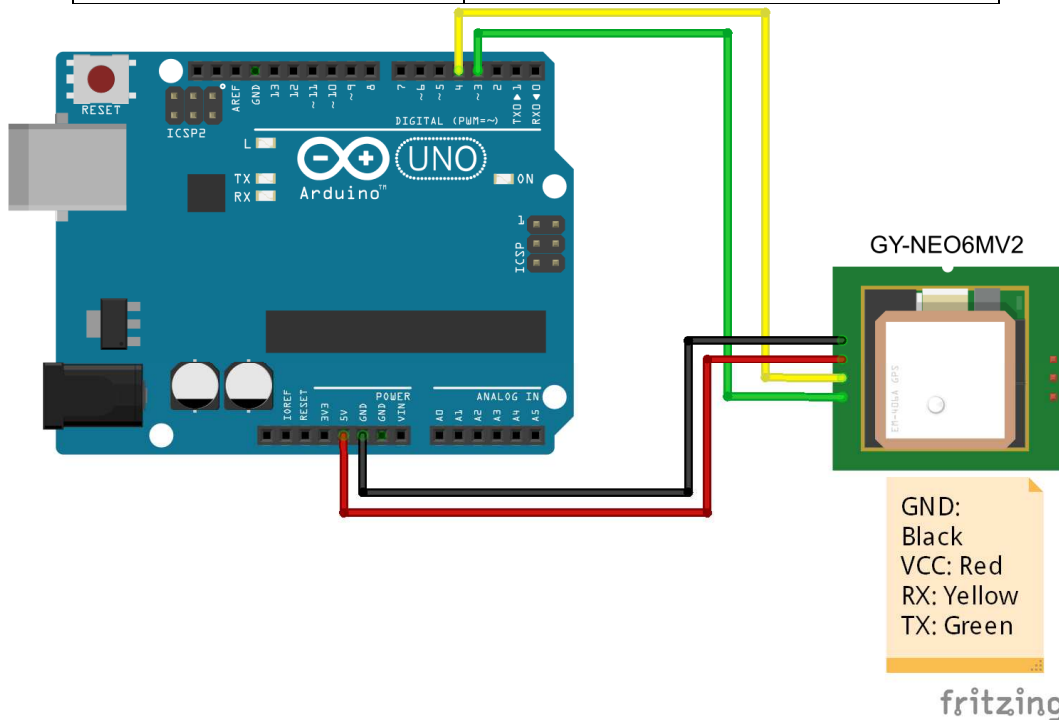


Figure IV. 5. Branchement de module GPS avec Arduino Uno.

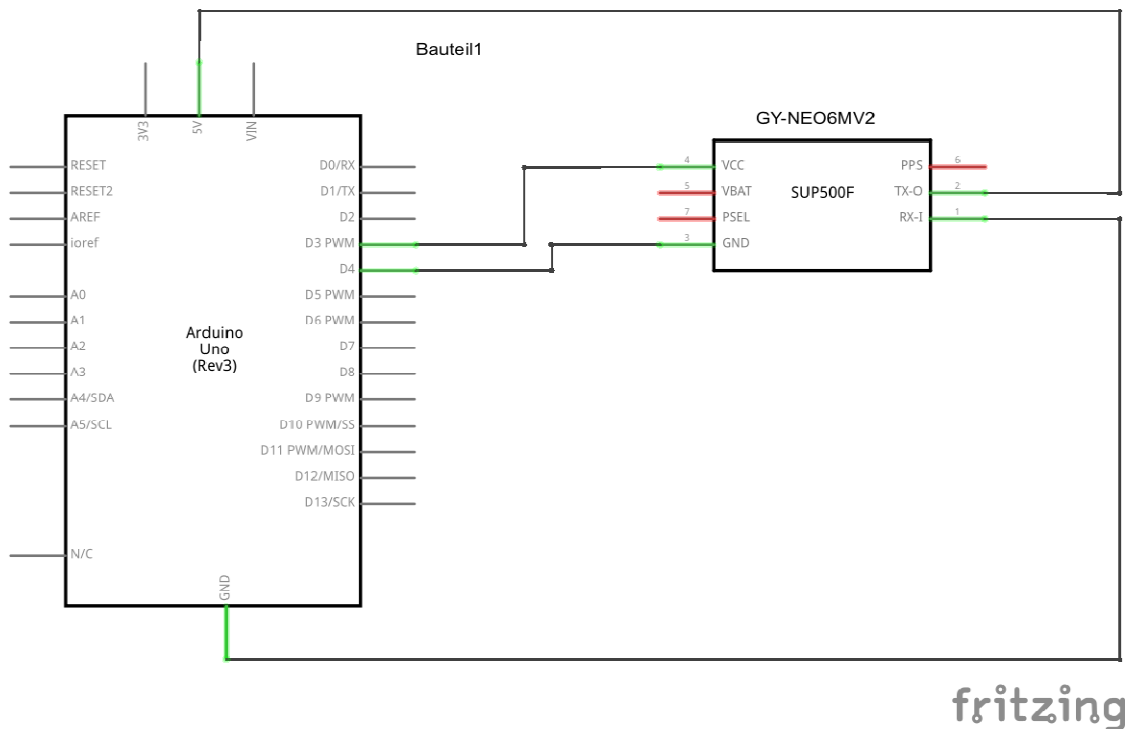


Figure IV. 6. Circuit électrique du module GPS avec la carte Arduino Uno.

IV.2.2.2. Programmation de Module GPS

Nous allons utiliser la bibliothèque `#include "TinyGPS++.h"` qui extrait automatiquement les coordonnées GPS à partir des trames NMEA.

La bibliothèque `#include "SoftwareSerial.h"` a été développée pour permettre la communication série sur d'autres broches numériques de l'Arduino de manière logicielle. Il est possible de gérer plusieurs ports séries logiciel avec des vitesses allant jusqu'à 155200 *bps* cependant un seul peut recevoir des données à la fois.

Lorsque l'on ouvre le moniteur série (loupe en haut à droite de logiciel IDE Arduino), on voit apparaître des trames de type NMEA.

- ✓ Toutes les trames commencent par le signe \$,
- ✓ Toutes les trames finissent par le symbole * + checksum sur 2 chiffres en hexadécimal. Le checksum permet de vérifier l'intégrité des données transmises,

Les trames qui nous intéressent pour la localisation sont les trames qui commencent par \$GPGGA. (GP =GPS et GGA=localisation).

Le module communique en 9600 *bauds* et envoie les informations de localisation toutes les 1s.

Le tableau IV.2 ci-dessous présente les commandes utilisés et leurs rôles dans la programmation d'une carte Arduino avec un module GPS. Dans la sous section suivante, on

donne un exemple d'un programme en utilisant un Arduino et le module GPS (GY-NEO6MV3).

Tableau IV. 2. Commandes et les rôles d'un programme GPS.

Commande	Rôle
#include<TinyGPS++>.h	Appel de la bibliothèque
TinyGPSPlusgps	Création de l'instance de l'objet (GPS)
Gps.encode(data)	Transmet les caractères reçus à la bibliothèque pour analyse
Gps.location.isUpdated()	Localisation mise à jour
Gps.location.lat()	Renvoie la latitude sous la forme d'un nombre de type double
Gps.location.lng()	Renvoie la longitude sous la forme d'un nombre de type double
Gps.altitude.meters()	Renvoie l'altitude en mètres sous la forme d'un nombre de type double
Gps.speed.kmph()	Renvoie la vitesse en <i>Km/h</i> sous la forme d'un nombre de type double
Gps.satellites.value()	Renvoie le nombre de satellites utilisés sous la forme d'un nombre de type unsigned long

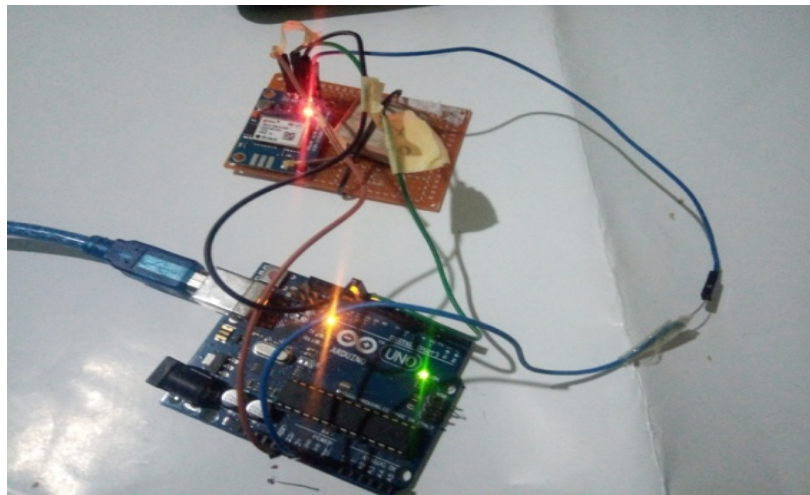
IV.2.2.3. Exemple de programme

```
#include "TinyGPS++.h"
#include "SoftwareSerial.h"
SoftwareSerial serial_connection(11, 10); //RX=pin 10, TX=pin 11
TinyGPSPlus gps; //This is the GPS object that will pretty much do all the gruntwork with the NMEA data
void setup() {
  Serial.begin(9600); //This opens up communications to the Serial monitor in the Arduino IDE
  serial_connection.begin(9600); //This opens up communications to the GPS
  Serial.println("GPS Start"); //Just show to the monitor that the sketch has started
}
void loop() {
  while(serial_connection.available()) //While there are characters to come from the GPS
  {
    gps.encode(serial_connection.read()); //This feeds the serial NMEA data into the library one char at a time
    if(gps.location.isUpdated()) //This will pretty much be fired all the time anyway but will at least reduce it to
    only after a package of NMEA data comes in
    //Get the latest info from the gps object which it derived from the data sent by the GPS unit
    Serial.println("Satellite Count:");
    Serial.println(gps.satellites.value());
    Serial.println("Latitude:");
    Serial.println(gps.location.lat(), 6);
    Serial.println("Longitude:");
    Serial.println(gps.location.lng(), 6);
    Serial.println("");
  }
}
```

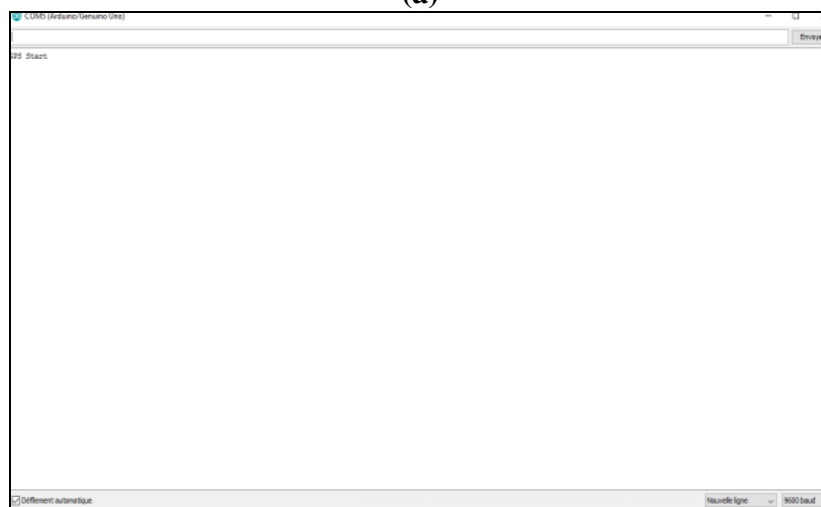
IV.2.2.4. Fonctionnement de montage GPS

La localisation indoor permet donc de savoir plus ou moins précisément la localisation d'un individu ou d'un objet dans un espace fermé. Le GPS fonctionne dans les zones extérieures par contre ne fonctionne pas dans les zones intérieures est limitée dû à la mauvaise pénétration des signaux GPS comme des bâtiments a cause des obstacles, par exemple les hôpitaux ou à l'intérieur des maisons.....etc. Les signaux des satellites faibles et difficiles à recevoir dans les endroits fermés. Dû à la présence d'obstacles entre les satellites et le récepteur. Dans ce cas, nous avons fait une expérience à l'intérieur et à l'extérieur d'un environnement, et les résultats étaient comme le montrent les figures suivantes.

Notre première expérience est dans le cas d'un environnement intérieur. La figure IV.7(a) représente notre montage a testé. D'après la figure IV.7(b), notre montage ne fonctionne pas correctement.



(a)



(b)

Figure IV. 7. (a) Montage de la localisation GPS, (b) affichage de moniteur série de l'arduino sur PC.

Par contre dans la deuxième expérience (outdoor), notre montage se fonctionne bien, comme la représente la figure suivante :

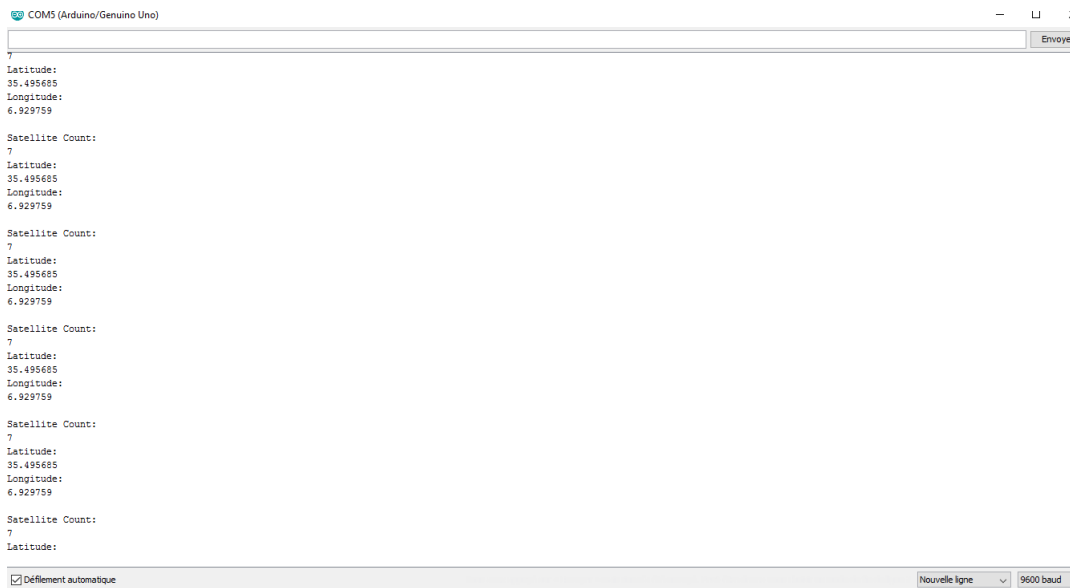


Figure IV. 8. Affichage de moniteur série de l'arduino sur PC, dans le cas d'un environnement extérieur.

Les informations de longitude et de latitude reçues ont été utilisées pour identifier notre emplacement sur Google Maps (voir Figure IV.9).

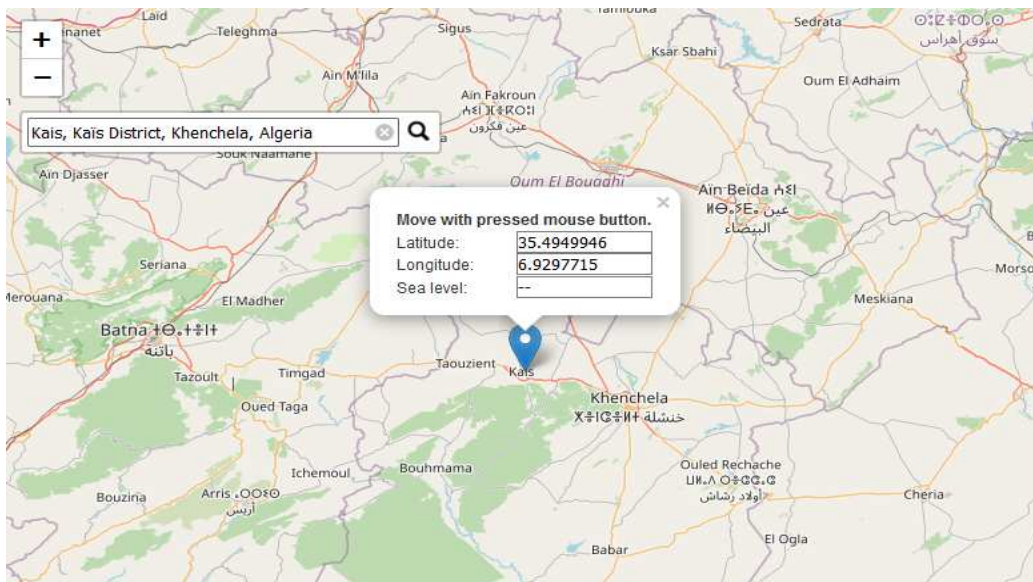


Figure IV. 9. La localisation outdoor d'un utilisateur à l'aide de Google Maps.

Dans le cas d'un environnement fermé, il est difficile de trouver la localisation exacte d'un utilisateur. Pour cela, nous avons proposée dans les sections suivantes, d'utiliser deux cartes basés sur la technologie WiFi: ESP32 et Raspberry Pi. L'objectif de ce travail est de créer et sauvegarder des valeurs de l'indicateur de puissance RSSI, puis utiliser ces valeurs dans les

modèles de l'intelligence artificielle développé dans le chapitre précédent pour faire la localisation indoor.

IV. 3. ESP32

L'ESP32 est une petite carte électronique, appelée microcontrôleur, facile à prendre en main grâce à ses ressemblances avec la carte Arduino qui est bien plus répandue. L'ESP32 peut avoir différentes tailles mais la version la plus courante est la suivante :

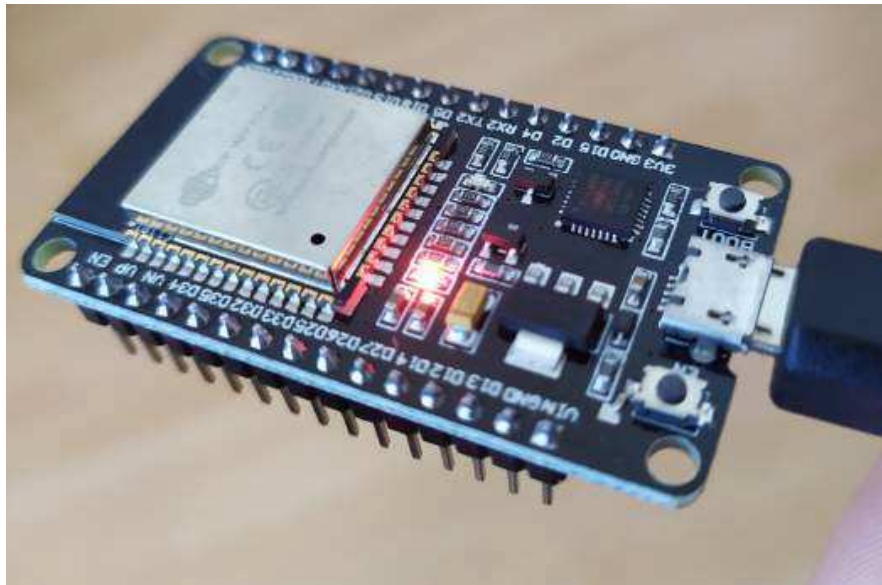


Figure IV. 10. La carte ESP32 DEVKIT WROOM

L'assignation des bornes du «ESP32 DEVKIT V1, est représenté dans la figure suivante :

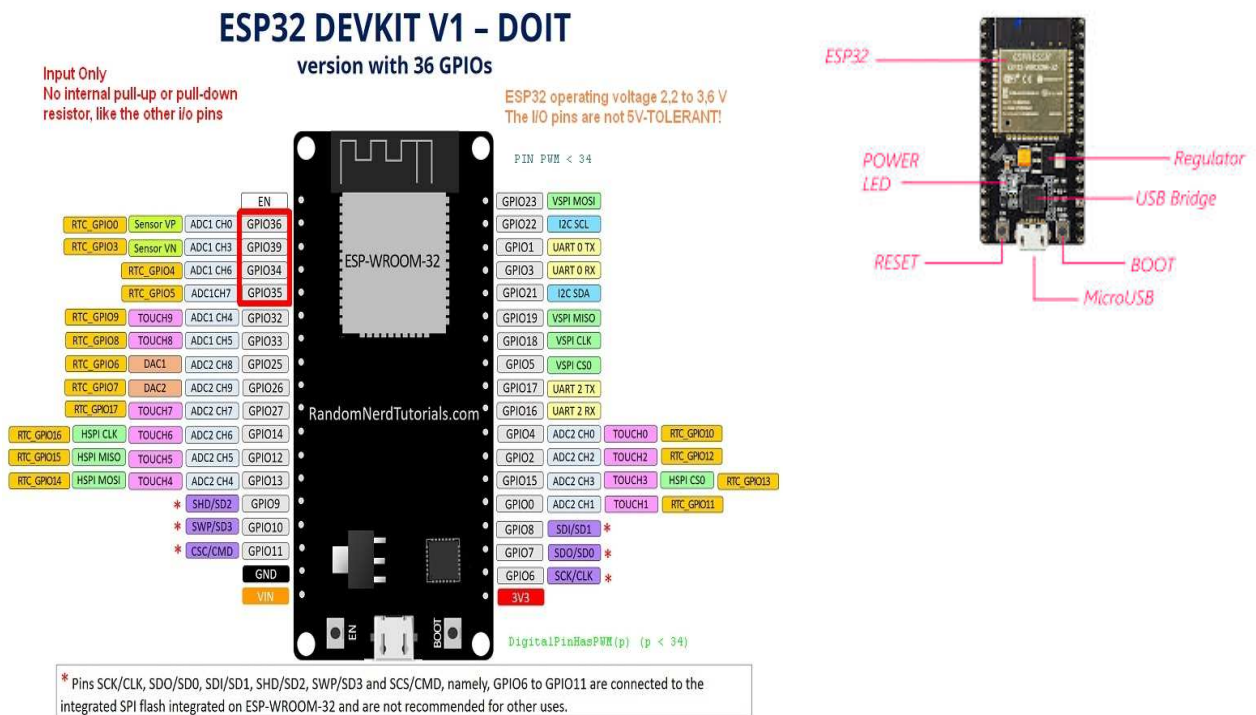
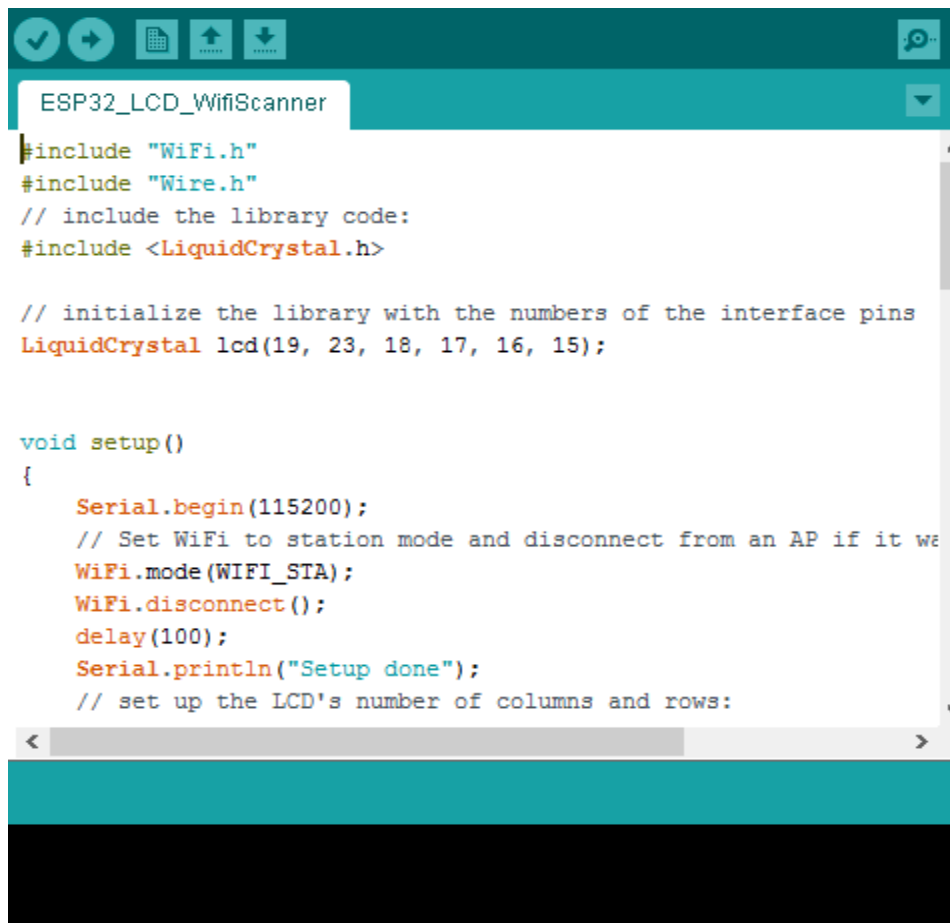


Figure IV. 11. Les Caractéristique de La carte ESP32 DEVKIT WROOM.

L'entrée Vin du «ESP32 DEVKIT DOIT» fonctionne entre 4,5 et 12 volts. Un régulateur (NCP1117) sur le circuit abaisse la tension à 3,3 volts et fournit un courant maximum de 800ma.

- Le ESP32 fonctionne entre 2,2 et 3,6 volts.
- Les broches d'entrées/sortie du «ESP32 DEVKIT DOIT» ne supportent pas le 5 volts. Le voltage à l'entrée ne doit pas dépasser 3,6 volts.
- Les GPIO 36, 39, 34 et 35 sont des broches d'entrée de signal seulement (I). Elles ne possèdent pas de résistances internes de pull-up et pull-down.
- Les broches GPIO < 34 peuvent fonctionner en PWM
- Le courant maximal absolu par GPIO est de 40 mA

Exemple de programme



```
ESP32_LCD_WifiScanner
#include "WiFi.h"
#include "Wire.h"
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(19, 23, 18, 17, 16, 15);

void setup()
{
  Serial.begin(115200);
  // Set WiFi to station mode and disconnect from an AP if it wa
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  Serial.println("Setup done");
  // set up the LCD's number of columns and rows:

```

IV.3.1. Fonctionnement de montage ESP32

Nous avons utilisé ESP32 avec l'affichage à cristaux liquides (LCD), pour l'objectif d'afficher et sauvegarder les valeurs de l'indicateur de puissance RSSI, et ensuite utilisé ces valeurs à l'entrée de notre modèle de classification SVM pour trouver la localisation exacte de l'utilisateur dans un environnement fermé.

Les Composants nécessaires pour réaliser le montage basé sur ESP32, sont représentés dans la figure suivante :

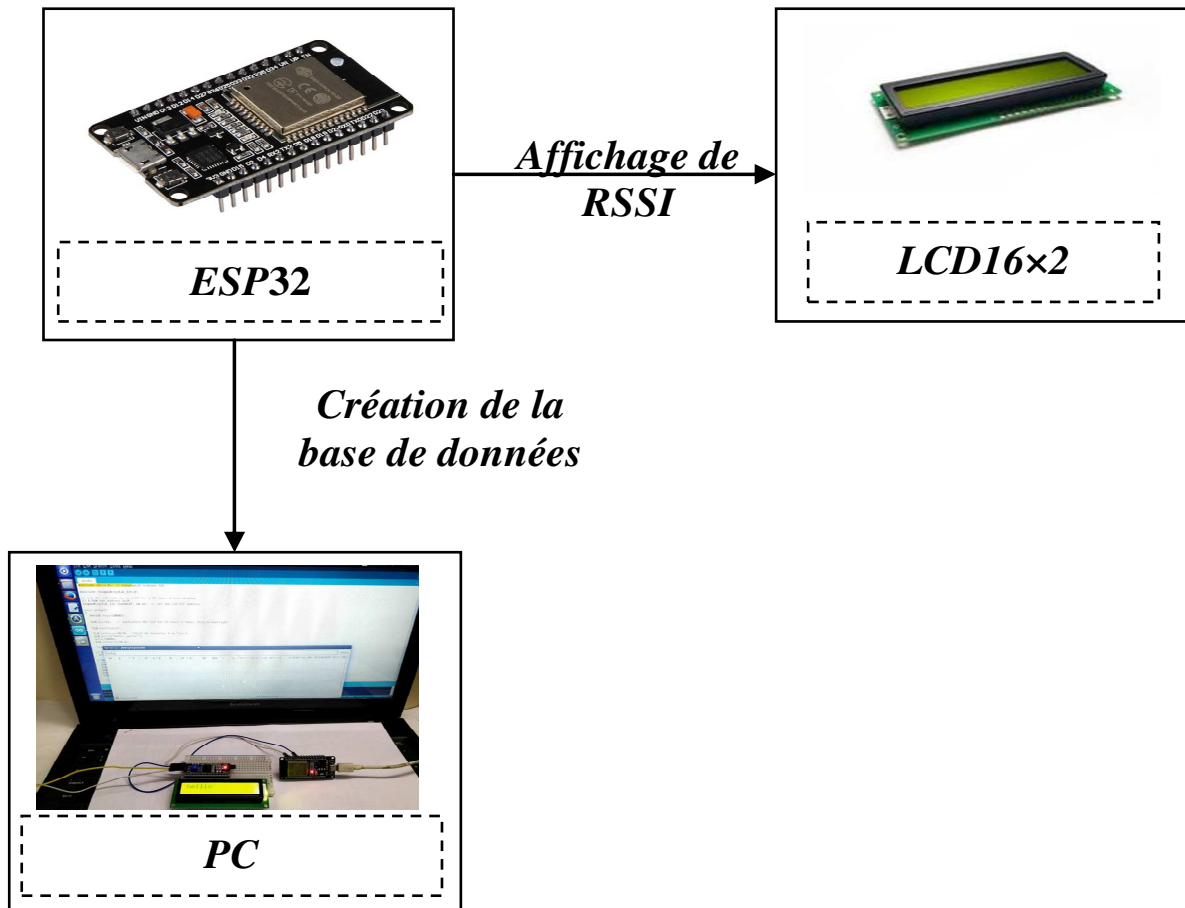


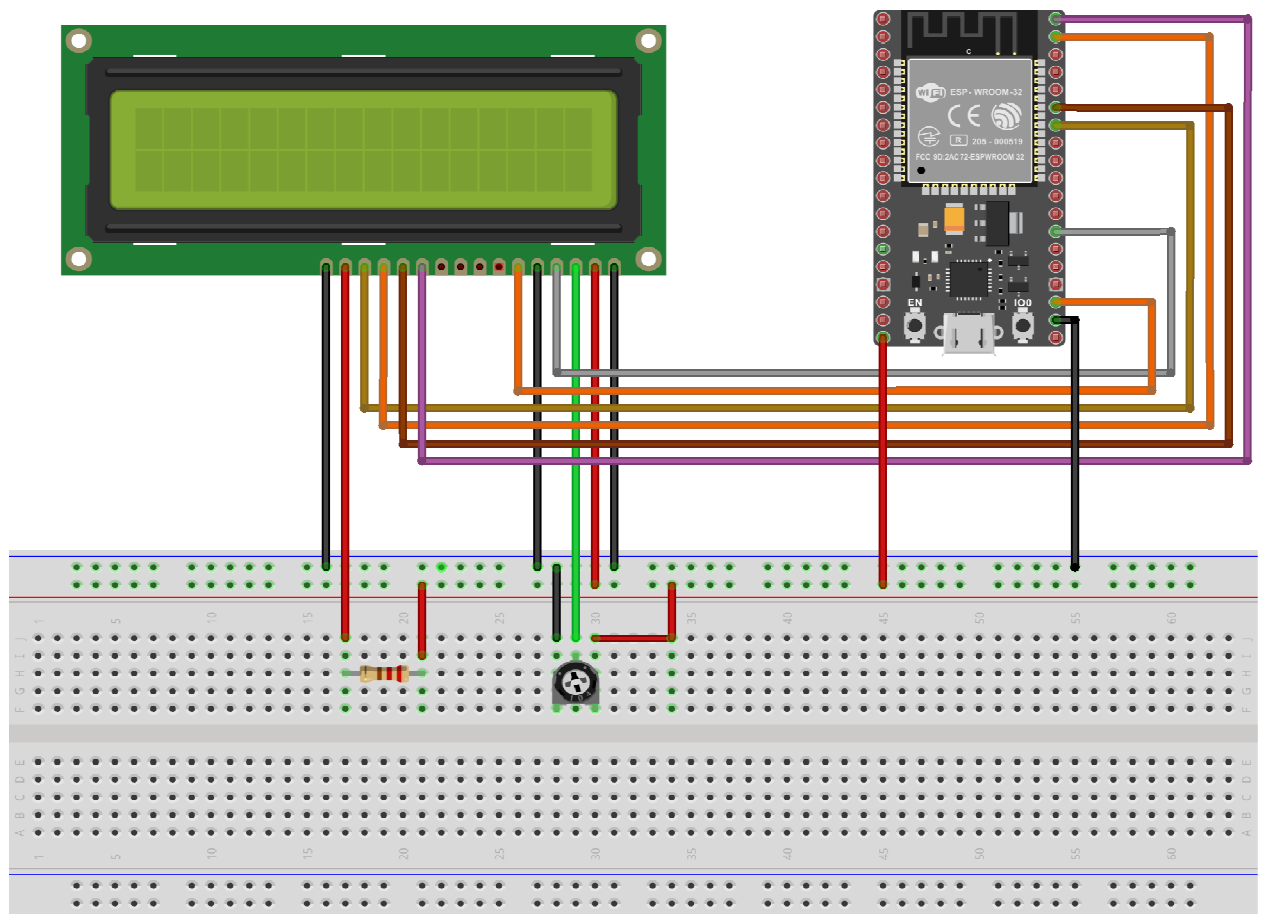
Figure IV. 12. Schéma récapitulatif du système de positionnement basé sur ESP32.

Dans ce projet nous avons utilisé un afficheur LCD, et la carte ESP32 pour afficher les valeurs de RSSI. Pour créer les bases de données, nous avons lié la carte ESP32 avec notre PC, et sauvegarder les mesures obtenues pour chaque emplacement, et par la suite utilisé cette base de données pour l'apprentissage de classifieur SVM.

Nous pouvons connecter les broches de communication de l'écran LCD à toutes les broches d'entrées/sorties de la carte ESP32. Dans ce projet, ils sont connectés comme indiqué dans le Tableau IV.3. Le branchement et le circuit électrique de l'écran LCD avec ESP32 sont représentés dans les Figures IV.13 et IV.14.

Tableau IV. 3. Connexions des broches LCD de communication.

Broches LCD	Broches ESP32
PIN01-VSS	-> GND
PIN02-VDD	5V
PIN03 V0	10K Pot (Middle pin)
PIN04 RS	GPIO19
PIN05 RW	GND
PIN06 E	GPIO23
PIN11 D4	GPIO18
PIN12 D5	GPIO17
PIN13 D6	GPIO16
PIN14 D7	GPIO15
PIN15 A	5V
PIN16 K	GND



fritzing

Figure IV. 13. Assemblage du ESP32WROOM et LCD 16x2.

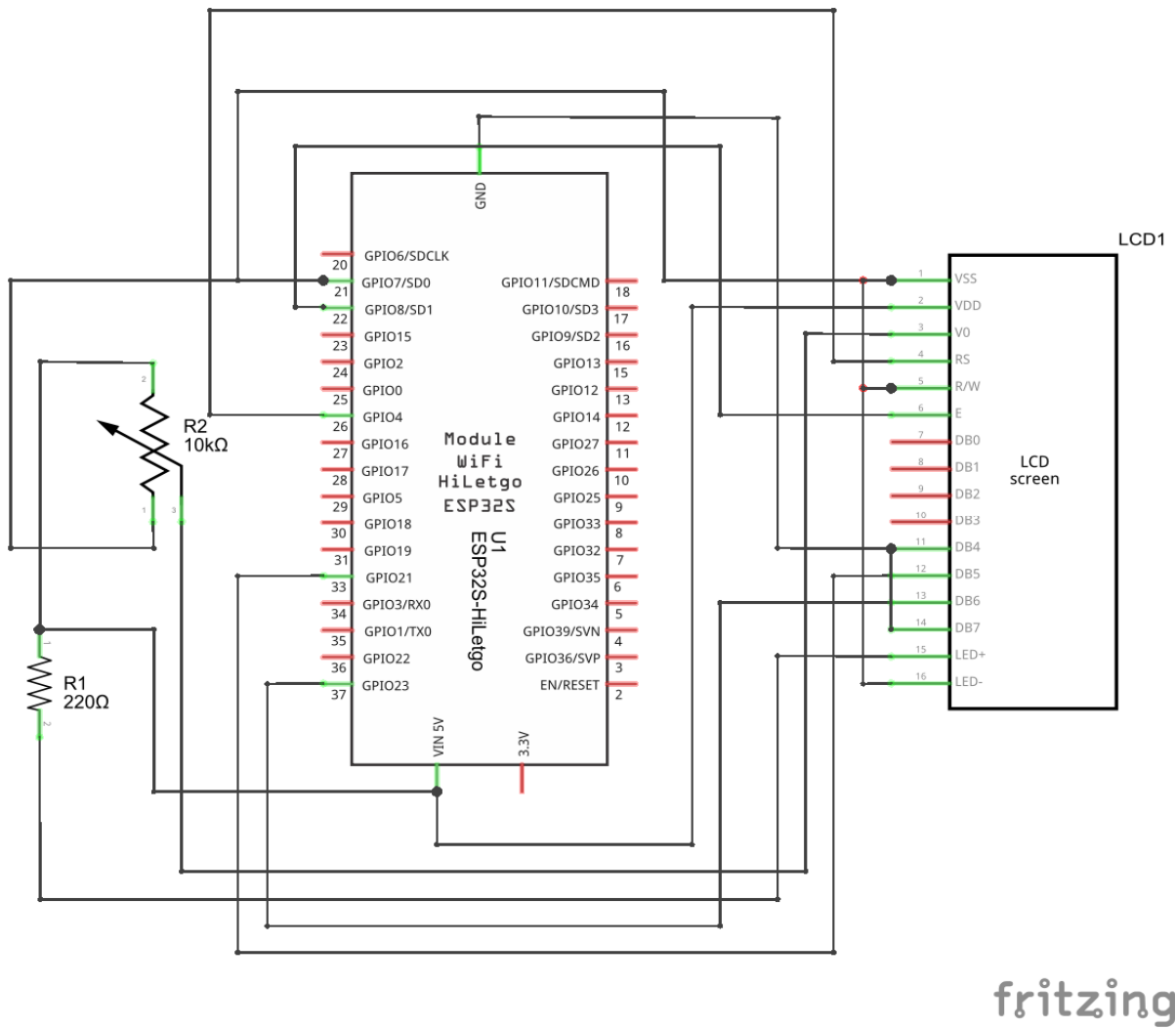


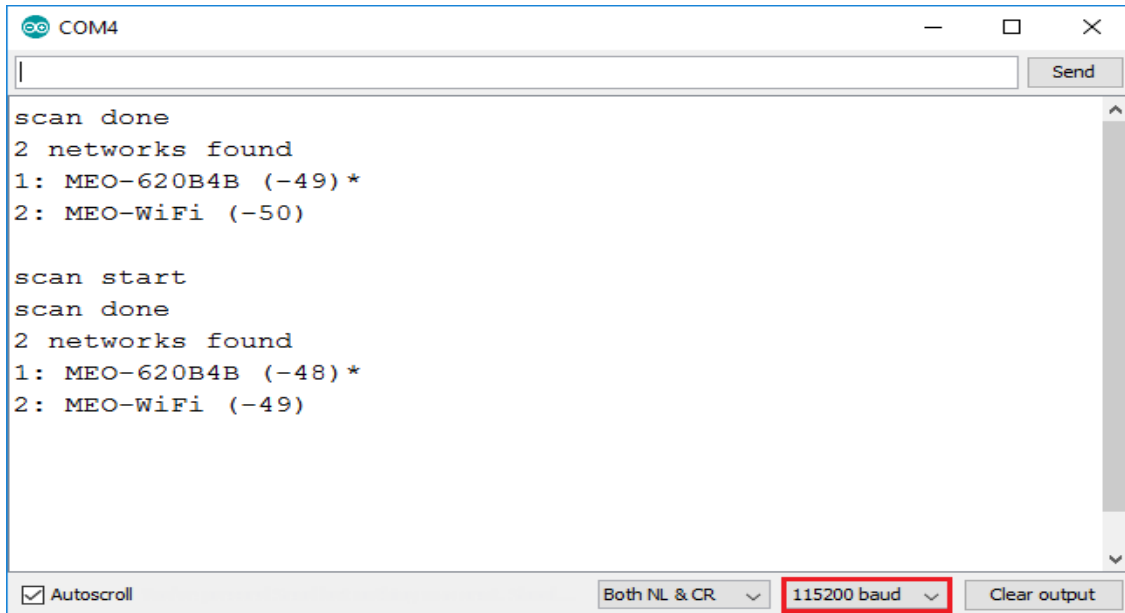
Figure IV. 14. Circuit électrique du module ESP32WROOM et LCD 16x2.



Figure IV. 15. Affichage de la qualité du signal (RSSI).

La figure IV.15 représente le schéma pratique du montage ESP32. Comme montre cette figure, notre objectif principal est de mesurer la qualité de signal WiFi (RSSI), et par la suite sauvegarder ces mesures pour l'utiliser ultérieurement.

La figure suivante représente, la qualité du signal RSSI pour les différents retours disponibles dans notre environnement.



```
COM4
scan done
2 networks found
1: MEO-620B4B (-49)*
2: MEO-WiFi (-50)

scan start
scan done
2 networks found
1: MEO-620B4B (-48)*
2: MEO-WiFi (-49)
```

The screenshot shows a terminal window titled 'COM4' with a 'Send' button. The output displays two scan results. The first scan shows two networks: 'MEO-620B4B' with an RSSI of -49 dBm and 'MEO-WiFi' with an RSSI of -50 dBm. The second scan shows the same two networks: 'MEO-620B4B' with an RSSI of -48 dBm and 'MEO-WiFi' with an RSSI of -49 dBm. At the bottom of the window, there are controls for 'Autoscroll' (checked), 'Both NL & CR' (dropdown), '115200 baud' (dropdown, highlighted with a red box), and 'Clear output'.

Figure IV. 16. RSSI des réseaux disponibles à proximité d'ESP32.

IV. 4. Raspberry Pi

Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (Matlab, python, scratch...). Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser le Raspberry Pi est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et des autres logiciels compatibles.

IV.4.1. Aspects Logiciel

Toutes les applications compatibles avec le système d'exploitation et le processeur ARM, ou utilisant un environnement d'exécution virtuel (Java, émulateurs...) sont susceptibles de fonctionner : Python, MATLAB, ... Les principales contraintes portent sur les performances du processeur et la mémoire vive disponible (256 Mo). Ce dernier point a toutefois été corrigé avec l'arrivée des versions embarquant 512 Mo de mémoire vive ce qui est suffisant pour lancer le processus du contrôle et le serveur web sur le système d'exploitation Raspbian.

La plupart des systèmes qui fonctionnent sur Raspberry *Pi* sont des versions du système D'exploitation Linux. Parce que Linux est open source, les développeurs peuvent l'adopter Pour des buts spécifiques. La distribution recommandée est Raspbian. C'est un système d'exploitations libres basées sur Debian optimisé pour le matériel Raspberry *Pi*. Cependant Raspbian fournit plus qu'un simple système d'exploitation : il est livré avec plus de 35.000 Paquets, des logiciels précompilés qui nous facilitent le développement soft. Il y a autre système d'exploitation crée par Microsoft spécialement pour le Windows 10 IoT qui peut être Installé sur notre carte Raspberry *Pi* 3.



Figure IV. 17. Logo du système d'exploitation Raspbian et Windows 10 IoT.

IV.4.2. Accès à distance au Raspberry *Pi*

On peut accéder à la carte Raspberry *Pi* via le protocole SSH et/ou VNC, si on connaît l'adresse IP de la carte.

IV.4.3. Choix de l'adresse IP du Raspberry *Pi*

Pour trouver l'adresse IP de notre Raspberry *Pi*, nous avons utilisé Advanced IP Scanner qui est un scanner de réseau local utilisé sous Windows. La figure suivante montre l'interface de cet outil et l'information fournie par ce dernier concernant le Raspberry *Pi* :

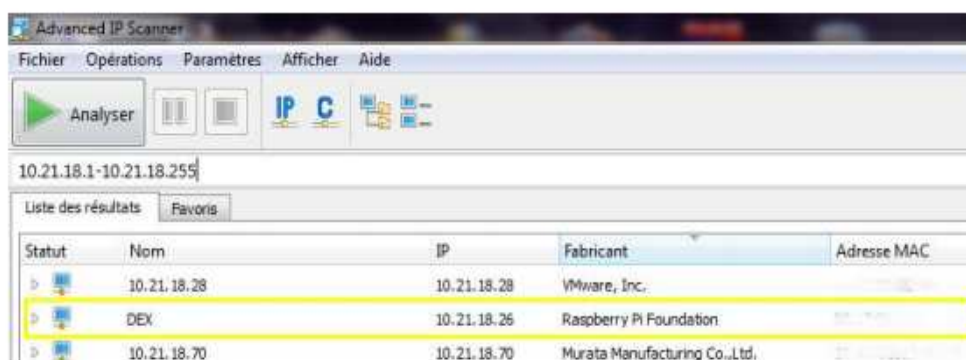


Figure IV. 18. Adresse IP de la carte Raspberry *Pi*.

Après que nous avons trouvé l'adresse IP, il faut la fixer afin de nous permettre l'accès à distance au Raspberry Pi, pour cela on va suivre les étapes suivantes :

Configuration réseau sous « Raspbian For Robots ». Par défaut, il contient les valeurs suivantes :

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet manual wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

La signification de chaque ligne de ce fichier est la suivante :

- auto lo : va démarrer l'interface automatiquement lors de la séquence de boot.
- Iface lo inet loopback : définition de l'interface loopback (interface virtuelle indispensable qui permet de supprimer les paquets erronés)
- iface eth0 inet dhcp : l'interface eth0 (le port RJ45 du Raspberry) sera configurée en envoyant une requête DHCP sur le réseau. Donc IP dynamique.

Les quatre lignes suivantes permettent de configurer le Wi-Fi via le fichier de configuration de wpa_supplicant.

Cette configuration permet de placer une adresse IP statique pour cela nous allons changer l'interface Ethernet (eth0) en mettant à jour la configuration de l'interface : IP, masque et passerelle comme indique la figure suivante :

```
root@dex:/etc# cd network/
root@dex:/etc/network# ls
if-down.d      if-pre-up.d  interfaces    run
if-post-down.d if-up.d      interfaces.bkp
root@dex:/etc/network# cat interfaces
auto lo

iface lo inet loopback
iface eth0 inet static
address 10.21.18.26
netmask 255.255.248.0
gateway 10.21.16.100
dns-nameservers 8.8.8.8

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface wlan0 inet dhcp
```

Figure IV. 19. Fichier de configuration réseau.

La ligne après gateway (passerelle) sert à définir les serveurs de noms de domaine (DNS) pour qu'on puisse se connecter à une ressource réseau externe (convertir un nom alphanumérique «par exemple `www. Google.com`» en une adresse réseau numérique «172.217.19.164»). Le DNS 8.8.8.8 est un DNS Google

Finalement le redémarrage de Raspberry *Pi*.

Alors à cette étape nous avons donné à notre carte une adresse IP fixe qui va nous servir à la manipulation de la carte à distance via le protocole SSH et ou VNC.

IV.4.4. Connexion à distance via SSH

Secure Shell est un protocole de réseau crypté pour initier des sessions Shell textuelles sur des machines distantes de manière sécurisée. Cela permet à un utilisateur d'exécuter des commandes sur l'invite de commande d'une machine sans qu'ils soient physiquement présents à proximité de la machine. SSH a été créé en 1995 pour le principal but est de permettre la prise de contrôle à distance d'une machine à travers une interface en lignes de commande.

Pour faire le contrôle à distance sur le Raspberry *Pi* on doit installer le SSH côté d'un serveur SSH sur notre Raspberry *Pi* et d'un autre d'un client SSH sur notre ordinateur, mais heureusement le serveur SSH est déjà installé et activé par défaut sur la Raspberry *Pi*, ce qui reste seulement d'installer un client SSH sur notre ordinateur ; pour cela on va installer le logiciel Putty sous Windows qui est un client SSH.



Figure IV. 20. Logiciel Putty.

Après son installation on entre l'adresse IP de la carte puis on clique sur «Open», et après l'authentification (en entrant le nom d'utilisateur pi et notre mot de passe), on se connectera directement sur l'invite des commandes de notre système d'exploitation Raspbian For robots comme l'indique la figure suivante:



```
pi@dex: ~  
login as: pi  
pi@10.21.18.26's password:  
Linux dex 4.1.20-v7+ #862 SMP Sat Mar 19 20:37:37 GMT 2016 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Apr 8 06:02:27 2016 from 10.21.19.88  
xhost: unable to open display ""  
pi@dex ~$
```

Figure IV. 21. Invite de commande via SSH.

IV.4.5. Caractéristique de la carte Raspberry Pi

- CPU: Processeur SoC de noyau de Cortex-A53 de quadruple BCM2837 64bit de Broadcom fonctionnant à 1.2GHz
- Mémoire: 1 Go de RAM LPDDR2
- 4 ports USB2.0 avec une sortie jusqu'à 1.2A
- En-tête GPIO étendu à 40 broches
- Sortie vidéo / audio via un connecteur 3,5 mm 4 broches, HDMI, caméra CSI ou LCD brut (DSI)
- Stockage: Support pour cartes micro-SD
- Port Ethernet 10/100 (RJ45)
- WiFi: 2,4 GHz, 802.11n (Broadcom BCM43438)
- Bluetooth 4.1 (Broadcom BCM43438)
- Périphériques de bas niveau:
 - 27 x GPIO
 - UART
 - Bus I2C
 - Bus SPI avec deux choix de puces

+ 3,3 V

+ 5V

GND

- Alimentation : 5 Vcc/maxi 2.5 A * via prise micro-USB (* intensité maxi si toutes les fonctions sont utilisées).
- Bus: SPI, I2C, série.

IV.4.6. Fonctionnement de montage à base de Raspberry Pi

Nous avons utilisé Raspberry Pi avec routeur WiFi, pour l'objectif de calcul le RSSI et afficher les valeurs de l'indicateur de puissance RSSI sur l'cran(PC), et ensuite utilisé ces valeurs à l'entrée de notre modèle de classification SVM pour trouver la localisation exacte de l'utilisateur dans un environnement fermé.

Les composants nécessaires pour réalisé le montage basé sur Raspberry Pi avec routeur WiFi, sont représentés dans les deux figures suivantes:

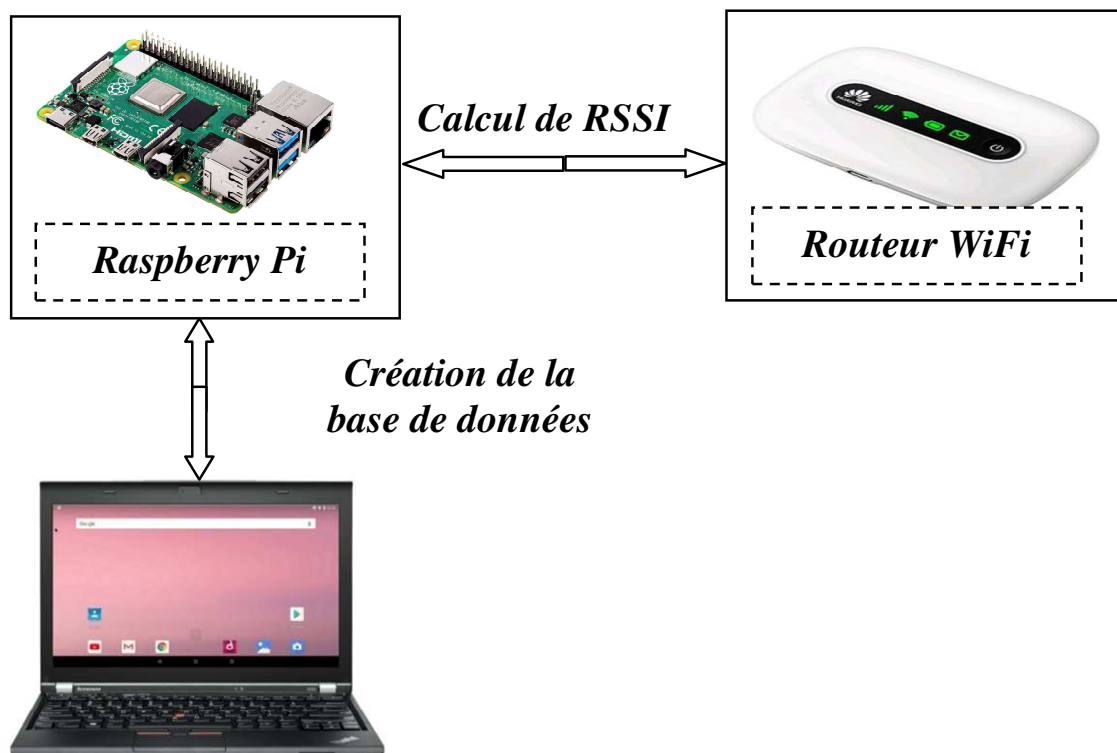


Figure IV. 22. Mesure de la qualité du signal (RSSI) à base de Raspberry Pi.

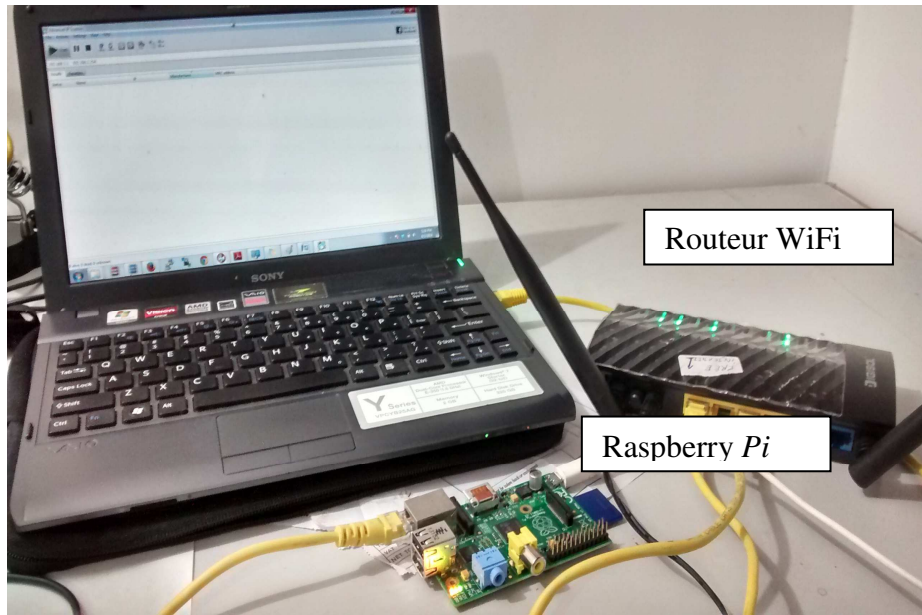


Figure IV. 23. Connectez Raspberry Pi au PC.

Comme on peut le voir sur la Figure IV.24, le terminal de Raspberry Pi balayer et capturer les indicateurs RSSI de tous les routeurs connectés. Donc le Raspberry Pi est toujours capable de fonctionner comme un nœud de capteur pour capturer confortablement le RSSI entrant.

L'avantage principal de la carte Raspberry Pi, est que contient un CPU pour faire des calculs et des opérations dures, donc on peut implémenter notre algorithme de classification SVM dans le Raspberry Pi pour la localisation en temps réel. Par contre la carte ESP32, on peut seulement calculer les indicateur RSSI et construire les bases de données.

```

pi@rasberrypi: ~
Quality=58/70 Signal level=-52 dBm
Quality=65/70 Signal level=-45 dBm
Quality=27/70 Signal level=-83 dBm
pi@rasberrypi:~$ iwlist wlan0 scanning|egrep "ESSID" && iwlist wlan0 scanning|
egrep "Signal"
ESSID:"Master_Telecom"
ESSID:""
ESSID:"dfouzi"
ESSID:"AndroidAPD4E4"
Quality=70/70 Signal level=-15 dBm
Quality=58/70 Signal level=-52 dBm
Quality=65/70 Signal level=-45 dBm
Quality=27/70 Signal level=-83 dBm
pi@rasberrypi:~$ iwlist wlan0 scanning|egrep "ESSID" && iwlist wlan0 scanning|
egrep "Signal"
ESSID:"Master_Telecom"
ESSID:""
ESSID:"dfouzi"
ESSID:"AndroidAPD4E4"
Quality=70/70 Signal level=-15 dBm
Quality=58/70 Signal level=-52 dBm
Quality=65/70 Signal level=-45 dBm
Quality=27/70 Signal level=-83 dBm
pi@rasberrypi:~$

```

Figure IV. 24. Capture d'écran des RSSI sur Raspberry Pi via un terminal.

IV. 5. Conclusion

Dans ce dernier chapitre, nous éclairons l'environnement de travail, et les composants électroniques qui ont été utilisés pour connaître l'emplacement d'un utilisateur. Nous avons fait trois expériences pour déterminer le positionnement, la première expérience est l'utilisation d'un GPS, nous avons donc remarqué que le GPS ne pouvait pas déterminer l'emplacement exacte dans un environnement fermé. Pour cela nous avons proposée d'utilisé des cartes contenant la technologie WiFi pour résoudre ce problème.

La deuxième et la troisième expérience basées sur la carte ESP32 et Raspberry *Pi* démontre l'efficacité et les performances de notre méthode. Enfin, on a fait plusieurs tests afin d'éclairer mieux le fonctionnement de notre projet.

Conclusion générale

Conclusion générale

Notre travail s'inscrit dans le contexte de la localisation des utilisateurs dans un environnement intérieur (indoor). L'utilisation des réseaux WiFi, conçus initialement pour répondre à des besoins de communication, s'est rapidement élargie pour intégrer des besoins de localisation. Pour ce faire, nous avons étudiés, la localisation en extérieur en utilisant le système de localisation GPS, ainsi les systèmes de localisation en utilisant la technologie WiFi en intérieur, en présentant leurs principes d'une façon générale, ensuite, une partie pratique avec l'utilisation des différents techniques mentionnés précédemment. Après une exposition de notre méthode proposée de la localisation basée sur la technologie WiFi et les méthodes de l'intelligence artificielle. Dans le cadre de notre travail, ce mémoire décrit en détail les différentes étapes menées pour aboutir à ce système de localisation, en le vérifiant avec des tests théorique et pratique qui répondent à la problématique posée dans ce mémoire. Les différents résultats des tests conduits ont été assez satisfaisants.

Pour cela, notre première contribution a portée sur l'utilisation des techniques d'intelligence artificielle (IA): réseaux de neurones MLP, RBFN et machine à vecteurs de support SVM pour l'objectif de la localisation d'un utilisateur dans un environnement intérieur «indoor». Le développement de cette approche a été fondé sur deux étapes. La première étape est la phase calibrage, l'objectif de cette phase est la proposition de l'architecture optimale pour les techniques MLP, RBFN et SVM; pour atteindre notre objectif, nous avons utilisé une base de données basée sur les mesures de la qualité du signal WiFi (l'indicateur de puissance RSSI). Dans la deuxième étape, nous avons testé la robustesse et la performance des techniques de l'intelligence artificielle développées (MLP, RBFN, SVM), afin de l'utiliser en temps réel.

Comme on l'a vu dans la partie simulation (chapitre III), nous avons fait plusieurs combinaisons avec IA sur l'ensemble de données RSSI, les résultats obtenus avec le classifieur SVM sont nettement meilleurs par rapport à ceux obtenus avec les réseaux de neurones MLP et RBFN en terme de taux de classification.

Notre deuxième contribution a porté sur la réalisation d'un système de localisation en temps réel en utilisant trois montages électroniques: GPS, ESP32, et Raspberry Pi.

Le premier montage est basé sur la carte Arduino et le module GPS, ce montage ne fonctionne pas dans les zones intérieures est limitée dû à la mauvaise pénétration des signaux GPS, par exemple les hôpitaux ou à l'intérieur des maisons.....etc. Les signaux des satellites faibles et difficiles à recevoir dans les endroits fermés. Dû à la présence d'obstacles entre les satellites et le récepteur. Donc, dans le cas d'un GPS, il est difficile de trouver la localisation exacte d'un utilisateur dans un environnement intérieur.

Pour cela, nous avons proposée d'utiliser deux montages électroniques équipées de la technologie WiFi : ESP32 et Raspberry Pi. L'objectif de ces deux montages est de crée est sauvegarder des valeurs de l'indicateur de puissance RSSI, puis utilisé ces valeurs dans les modèles de l'intelligence artificielle développé dans la phase de calibrage pour faire la localisation indoor. L'avantage principal de la carte Raspberry Pi par rapport le circuit ESP32, est que contient un CPU pour faire des calculs et des opérations dures, donc on peut implémenter notre algorithme de classification SVM dans le Raspberry Pi pour la localisation en temps réel. Par contre la carte ESP32, on peut seulement calculer les indicateur RSSI et construire les bases de données.

Enfin, ce projet va nous permettre de développer de nombreux domaines du quotidien, tel que système d'aide à la navigation destiné aux malvoyants à accomplir leurs tâches de vie quotidiennes telles que les achats et les déplacements au sein des différents services (administration, hôpitaux, ...). Ce projet permet aussi la localisation des utilisateurs dans les systèmes domestiques intelligents (smart home systems), la localisation des criminels dans des régions fermées, l'obtention du nombre d'utilisateurs dans un point d'accès, etc.

Perspectives :

- ✚ L'intégration entre les multi-technologies, comme WiFi, RFID, GSM,...etc. La fusion d'informations de localisation provenant de différentes technologies aux niveaux signal, localisation ou propositionnel, n'est qu'une première étape vers la gestion intégrée d'information de localisation qui rendra possible le développement d'applications attentives au contexte dans les environnements intérieur (intra-bâtiments).
- ✚ Système hybride entre la localisation WiFi et les systèmes de suivi visuel, avec la puissance de calcul actuelle, il est possible de construire un système de suivi visuel à une fréquence vidéo qui peut suivre de multiples cibles en utilisant simplement des processus de détection au niveau pixel.
- ✚ La localisation des utilisateurs dans les systèmes domestiques intelligents à l'aide de l'internet des objets (IoT).
- ✚ Utilisation des protocoles de routage sur les réseaux Ad-Hoc.
- ✚ Etude approfondie sur les méthodes de classification:
 - Utilisation d'autre méthode pour déterminer les meilleurs caractéristiques de la qualité du signal WiFi.
 - Utilisation d'autres techniques d'intelligence artificielle pour la localisation automatique des utilisateurs : Apprentissage profond (Deep Learning).

Bibliographie

Bibliographie

- [1] N. E. H. Seghier, "Localisation d'un mobile dans un réseau UMTS," *Magister en électronique, Option: Technique de communication modernes, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf*, 2013.
- [2] Y. Messai, "Système de localisation «indoor» en utilisant des techniques intelligentes à base de traitement de l'image," *Mémoire de Master, Option: Systèmes des télécommunications, Université Abbes Laghrour, Khenchela*, 2020.
- [3] M. Elgandy, T. Guzsvinecz, C. Sik-Lanyi, "Identification of Markers in Challenging Conditions for People with Visual Impairment Using Convolutional Neural Network," *Applied Sciences*, vol. 9, pp. 1-24, 2019.
- [4] R. Tapu, B. Mocanu, T. Zaharia, "Wearable assistive devices for visually impaired: A state of the art survey," *Pattern Recognition Letters*, vol. 137, pp. 37-52, 2020.
- [5] E. Valero, A. Adán, C. Cerrada, "Evolution of RFID Applications in Construction: A Literature Review," *Sensors*, vol. 15, pp. 15988–16008, 2015.
- [6] M. Anne, J. L. Crowley, V. Devin, G. Privat, "Localisation intra-bâtiment multi-technologies: RFID, Wifi et vision," *Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, pp. 29-35, 2005.
- [7] N. Benoudjit, K. Ferroudji, M. Bahaz, A. Bouakaz, "In vitro microemboli classification using neural network models and RF signals," *Ultrasonics*, vol. 51, pp. 247-252, 2011.
- [8] N. Benoudjit, M. Verleysen, "On the kernel widths in radial-basis function networks," *Neural Processing Letters*, vol. 18, pp. 139-154, 2003.
- [9] V. N. Vapnik, "Statistical Learning Theory," *Edition Wiley*, 1998.
- [10] F. Melgani, L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, pp. 1778-1790, 2004.
- [11] S. Belkhier, L. Hammiche, "Localisation de mobiles en intérieur avec WIFI," *Mémoire de Master Professionnel En Informatique, Option: Administration et Sécurité des Réseaux Informatiques, Université A/Mira de Bejaia*, 2015.
- [12] N. Mekhfi, N. Lahbab, "Mise au point d'une application de Géolocalisation Indoor via WiFi," 2018.
- [13] I. Amundson, X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," *International workshop on mobile entity localization and tracking in GPS-less environments, Springer, Berlin, Heidelberg*, pp. 235-254, 2009.
- [14] C. Fritsche, A. Klein, "On the performance of hybrid GPS/GSM mobile terminal tracking," *IEEE International Conference on Communications Workshops*, 2009.
- [15] A. Theljeoui, "Système hybride de localisation des personnes âgées dans un habitat intelligent," *Thèse de doctorat, Université Toulouse, Spécialité: Réseaux, Télécoms, Systèmes et Architecture*, 2017.
- [16] J. Trojan, J. A. Gaillard, J. Uriel, "Localisation de l'alphafoetoprotéine dans les dérivés neuroépithéliaux des tératocarcinomes de la souris," *Ann Pathol* vol. 3, pp. 137-145, 1983.
- [17] (2010). *US Air Force official website, Global Positioning System*, <http://www.af.mil/information/factsheets/factsheet.asp?id=119>.
- [18] S. Lejeune, "Les effets ionosphériques affectant les systèmes de positionnement par satellites," *Bulletin de la Société géographique de Liège*, vol. 47, pp. 25-31, 2006.
- [19] L. Arouche, N. Mezhoud, "Etude de la géolocalisation dans le réseau GSM," *Mémoire de Master, Option : Télécommunication, Université abderrahmane mira béjaia*, 2012.
- [20] A. Belabdelli, M. Oukaz, "Dimensionnement d'un réseau sans fil Wifi," *Ingénieur d'Etat en Télécommunications, Option : systèmes des télécommunications, Université Abou Baker Belkaide, Telmcen*, 2012.
- [21] G. LeGrand, A. Hecker, F. Springinsfeld, "Architecture flexible de réseau sans fil WiFi sécurisé," *Proc. SAR (Conference on Security and Network Architectures), La Londe, France*, 2004.

- [22] G. Dreyfus, J. M. Martinez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Héroult, "Réseaux de neurones," 39, vol. Paris: Eyrolles, 2002.
- [23] C. Touzet, "les réseaux de neurones artificiels, introduction au connexionnisme," *EC2*, 1992.
- [24] M. Parizeau, "Réseaux de Neurones GIF-21140 et GIF-64326," *Université de Laval*, 2004.
- [25] M. Fliess, "Automatique et corps différentiels," *Forum Math*, vol. 1, pp. 229-238, 1989.
- [26] B. Wierenga, J. Kluytmans, "Neural nets versus marketing models in time series analysis: A simulation study," *Marketing: Its Dynamics and Challenges*, vol. 1139 p. 1153, 1994.
- [27] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, pp. 251-257, 1991.
- [28] P. Borne, M. Benrejeb, J. Haggège, "Les réseaux de neurones: présentation et applications," *Editions Ophrys*, vol. 15, 2007.
- [29] M. Parizeau, "Réseaux de neurones," *Thèse de doctorat. Université LAVAL*, 2004.
- [30] M. Altet, "Les pédagogies de l'apprentissage," *Puf*, 2013.
- [31] R. A. Zemouri, D. Racoceanu, N. Zerhouni, "Réseaux de neurones récurrents à fonctions de base radiales: RRFR Application au pronostic," *Revue d'intelligence artificielle*, vol. 16, pp. 307-338, 2002.
- [32] S. Haykin, "Adaptive filters," *Signal Processing Magazine*, vol. 6, 1999.
- [33] F. Douak, N. Benoudjit, F. Melgani, "A two-stage regression approach for spectroscopic quantitative analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 109, pp. 34-41, 2011.
- [34] I. Wedding, K. Donald, J. C. Krzysztof, "Time series forecasting by combining RBF networks, certainty factors, and the Box-Jenkins model," *Neurocomputing* vol. 10, pp. 149-168, 1996.
- [35] N. K. Kasabov, "Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering," *MIT Press*, 1998.
- [36] V. Vapnik, "The nature of statistical learning theory," *Springer science & business media*, 2013.
- [37] B. Marir, M. Kalla, F. Douak, A. Daamouche, "A modular support vector machine for active learning of urban remote sensing images classification in Algeria," *Journal of the Indian Society of Remote Sensing*, vol. 46, pp. 515-529, 2018.
- [38] F. Douak, A. Tafast, D. Fouan, K. Ferroudji, A. Bouakaz, N. Benoudjit, "A wavelet optimization approach for microemboli classification using RF signals," *IEEE International Ultrasonics Symposium (IUS)*, pp. 1-4, 2016.
- [39] F. M. Zaiz, "Les supports vecteurs machines (SVM) pour la reconnaissance des caractères manuscrits arabes," *Magister en Informatique, Option : Intelligence artificielle et systèmes distribués, Université Mohamed Khider, Biskra*, 2010.
- [40] O. Bousquet, "Introduction au Support Vector Machines (SVM)" *Centre de mathématiques appliquées, Ecole polytechnique de Palaiseau*, 2001.
- [41] J. Shawe-Taylor, N. Cristianini, "Support Vector Machines and other kernel based learning methods," *Cambridge University Press*, 2000.
- [42] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, "Choosing kernel parameters for support vector machines," *Machine Learning*, vol. 46, pp. 131-160, 2001.
- [43] T. Joachims, "Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning," *MIT Press Cambridge, MA, USA*, 1999.
- [44] R. Rakotomalala, "SVM Support Vector Machine," *Supervised Learning-Classification, Université Lumière Lyon 2*, 2017.
- [45] H. Mebarki, O. Ghedir, "Système de communication entre les véhicules (VNET) en utilisant la communication sans fil et les méthodes de l'intelligence artificielle," *Mémoire de Master, Université Abbes Laghrour, Khenchela*, 2020.
- [46] A. Lendasse, W. Vincent, M. Verleysen, "Model selection with cross-validations and bootstraps-application to time series prediction with RBFN models," *Artificial Neural Networks and Neural Information Processing-ICANN/ICONIP, Springer, Berlin, Heidelberg*, pp. 573-580, 2003.

- [47] A. Djeflal, "Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données," *These de doctorat, Spécialité: Informatique, Université Mohamed Khider-Biskra*, 2012.
- [48] S. Knerr, L. Personnaz, J. Dreyfus, "Single-layer ll learning revisited : A stepwise procedure for building and training a neural network," *Optimization Methods and Software*, vol. 1, pp. 23-34, 1990.
- [49] J. C. Platt, N. Cristianini, J. Shawe-Taylor, "Advances in neural information processing systems, chapter Large margin DAGs for multiclass classification," *MIT Press*, vol. 12, pp. 547–553, 2000.
- [50] J. G. Rohra, B. Perumal, S. J. Narayanan, P. Thakur, R. B. Bhatt, "User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks," *Proceedings of Sixth International Conference on Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing*, vol. 546, pp. 286-295, 2017.
- [51] A. Abidi, S. Zeroual, "Etude et conception d'un détecteur d'objet en utilisant la communication sans fil, équipé d'un Raspberry Pi et une caméra," *Mémoire de Master, Option : Systemes de Télécommunications, Université Abbes Laghrour Khenchela*, 2019.