



**HAL**  
open science

## Empreinte radio BLE pour la localisation en intérieur

Maria Molina, Iness Ahriz Roula, Charlie Galle, Michel Terré

► **To cite this version:**

Maria Molina, Iness Ahriz Roula, Charlie Galle, Michel Terré. Empreinte radio BLE pour la localisation en intérieur. GRETSI 2022, Sep 2022, Nancy, France. hal-03956848

**HAL Id: hal-03956848**

**<https://hal.science/hal-03956848v1>**

Submitted on 25 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Empreinte radio BLE pour la localisation en intérieur

Maria Camila MOLINA<sup>1,2</sup>, Iness AHRIZ<sup>1</sup>, Charlie GALLE<sup>2</sup>, Michel TERRÉ<sup>1</sup>

<sup>1</sup>CEDRIC, Conservatoire National des Arts et Métiers  
292 Rue Saint-Martin, Cedex 03, 75141 Paris, France

<sup>2</sup>N-Vibe  
7 Avenue Parmentier, 75011 Paris, France

maria.molina@lecnam.net, iness.ahriz@lecnam.net  
charlie@n-vibe.fr, michel.terre@lecnam.net

**Résumé** – Cet article présente une étude comparative entre une architecture de localisation simple et une architecture hiérarchisée. Les solutions étudiées ont pour objectif une localisation intérieure basée sur l’empreinte RSS (Received Signal Strength) de la technologie BLE (Bluetooth Low Energy) en utilisant l’apprentissage automatique. Deux algorithmes ont été implémentés pour chaque stratégie, il s’agit du K plus proches voisins et le perceptron multi couches. Des expériences ont été menées dans un environnement intérieur montrant la faisabilité des stratégies étudiées et leurs performances similaires en termes de précision de localisation ne dépassant pas les 2 mètres d’erreur en moyenne.

**Abstract** – This article presents a comparative study of a simple localisation architecture and a hierarchical architecture. The studied solutions aim to localize a user in an indoor setting using RSS (Received Signal Strength) fingerprints of BLE (Bluetooth Low Energy) by using machine learning techniques. Two algorithms were implemented for each of the strategies, the K nearest neighbour algorithm and a multi-layer perceptron. Experimentation was carried out in an indoor environment showcasing the feasibility of the evaluated strategies as well as their similar performance regarding localization accuracy which stays within a 2-meter range on average.

## 1 Introduction

Grâce à l’émergence croissante de l’Internet des Objets et à l’importance de l’information de position dans ce contexte, la recherche sur ce thème attire l’attention dans la communauté des chercheurs. Les services basés sur la localisation (LBS) sont devenus un élément clé pour une myriade d’applications de pointe dans le domaine de l’Internet des objets (IoT) [1, 2, 3]. Pour n’en citer que quelques-uns, le traçage et le suivi du personnel et des marchandises en temps réel pour faciliter les opérations dans les entreprises, le marketing de proximité, la question du suivi des intrus dans le cas d’installations sensibles comme les systèmes bancaires, des applications dédiées aux personnes malvoyantes afin d’assurer des services de navigation autonomes et exacts dans un environnement complexe. Afin de répondre aux besoins pressants de ces activités, le point commun fondamental de ces applications LBS est l’efficacité et la précision de localisation.

La localisation dans les milieux extérieurs est assurée par le GPS (Global Positioning System) qui assure une précision de 5 mètres. L’absence de ces signaux dans les milieux intérieurs rend l’utilisation d’une telle solution impossible et plusieurs techniques de localisation ont été proposées. Elles se basent sur l’utilisation des signaux radio échangés par les objets connectés. Les méthodes existantes se basent sur des approches de cartographie géométriques ou sur l’empreinte radio (fingerprinting). Pour la cartographie géométrique, des paramètres spatiaux in-

termédiaires comme la distance ou la direction sont d’abord dérivés de certaines mesures physiques. Les paramètres typiques comprennent le temps de vol (Time of Flight) [4] et l’angle d’arrivée (Angle of Arrival) [5]. La position de la cible peut ensuite être déduite en utilisant des algorithmes de trilatération ou de triangulation. La performance de l’approche de cartographie géométrique dépend fortement de la condition de visibilité directe entre l’émetteur et le récepteur (Line of Sight). Les méthodes basées sur l’empreinte radio sont organisées en deux phases. Dans une phase hors ligne, les signatures sont collectées au niveau d’un ensemble de points de référence géolocalisés dans la zone d’intérêt pour construire la base de données d’empreintes radio. Pendant la phase en ligne, la signature mesurée à une position inconnue est comparée à la base de données construite hors ligne afin d’obtenir la localisation la mieux adaptée. Souvent, l’utilisation des données RSS est favorisée car c’est une information qui existe dans toutes les normes de communication sans fil et aucune action supplémentaire n’est nécessaire. Plusieurs travaux illustrent l’efficacité de modèles de machine learning traditionnels tels que les SVM (Support Vector Machine) [6] ou l’algorithme des K plus proches voisins [7] pour l’estimation de la position. Plus récemment, des techniques de fingerprint utilisant des réseaux de neurones profonds ont également montré de bonnes précisions dans le contexte de la localisation en intérieur [8, 9].

En s’inspirant de ces travaux, nous proposons de comparer une architecture de fingerprint simple avec une architec-

ture hiérarchisée dans laquelle la problématique de localisation est découpée en sous-problématiques plus simples à prendre en charge par un algorithme de machine learning. La localisation consistera à faire une identification de l'étage dans un bâtiment, puis d'une zone dans l'étage et ensuite d'une position précise à l'intérieur de cette zone. Ainsi, la localisation se fera avec différents niveaux de précision et l'utilisateur pourra choisir celui qui correspond le mieux à son application.

La suite de l'article sera organisée comme suit : la section 2 sera consacrée à la description de l'environnement expérimental. Dans la section 3 les méthodes et algorithmes mis en œuvre seront présentés. Les résultats seront présentés et discutés dans la section 4 et la section 5 servira de conclusion à l'article.

## 2 Présentation de l'environnement

L'environnement d'expérimentation est un immeuble de 5 étages muni d'un ascenseur. Ce dernier a une superficie totale d'environ 1000 m<sup>2</sup>. Nous nous concentrons dans cette étude sur une superficie de 177 m<sup>2</sup>. Chaque étage a été divisé en une ou plusieurs zones et sur chaque zone, des points repères ont été choisis. Au total, 14 zones et 60 points repères sont définis sur tout le bâtiment. Ce dernier est équipé de 25 balises Bluetooth Low Energy présentées par des points bleus sur la figure 1.

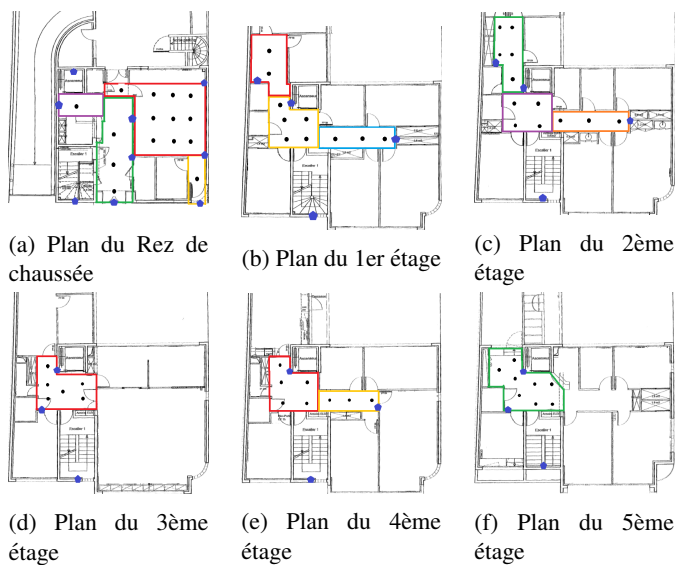


FIGURE 1 – Disposition des balises, zones, et points repères

Afin d'établir des jeux de données répertoriant toutes les zones et tous les points repères, des mesures RSS des 25 balises bluetooth ont été récupérées à l'aide d'un smartphone iPhone 11. Dans le cas des données correspondant aux zones, les enregistrements des valeurs RSS ont été effectués en se déplaçant pendant 4 minutes à l'intérieur de chaque zone définie. Dans le cas des données correspondant aux points repères, deux enregistrements de 2 minutes ont été effectués par point repère. Les deux enregistrements sur un même point repère ont été effectués dans des directions opposées.

L'application d'enregistrement des valeurs RSS permet d'en-

registrer les valeurs de toutes les 25 balises une fois par seconde. Ainsi, les données des points repères récoltées permettent de définir un jeu de données de 25 balises et 14400 observations. Les données récoltées pour la classification de zones permettent de définir un jeu de données de 25 balises et 3360 observations.

Dans le but de tester la performance des modèles entraînés et simuler le fonctionnement envisagé du système de localisation en intérieur, 10 trajets test ont été effectués. Les données RSS des 25 balises ont été enregistrées pendant ces trajets. Parmi les trajets test, 6 des trajets sont des trajets à plusieurs étages et 4 sont des trajets effectués sur un seul étage.

Les données ont été enregistrées pendant une période d'utilisation normale du bâtiment, avec des personnes en mouvement, l'ascenseur en fonctionnement régulièrement, l'ouverture et fermeture de portes ainsi que le déplacement de mobilier.

## 3 Méthodes

### 3.1 Architectures de fingerprint

#### 3.1.1 Fingerprint simple

Nous proposons, dans le cadre de la localisation en intérieur, l'utilisation de balises bluetooth à basse consommation couplées à des algorithmes exploitant les données RSS de ces dernières. Des algorithmes d'apprentissage statistique tels que l'algorithme des K plus proches voisins et le perceptron multicouches sont prévus à cet effet en tant que modèles dits de "fingerprint", permettant de pallier le manque de modèles de propagation connus. Ainsi, en établissant une base de données répertoriant différentes positions et les valeurs RSS associées à ces positions, il est possible de donner une identité unique à chaque position. En utilisant des algorithmes de classification, il est possible d'enregistrer les valeurs RSS des différentes balises à une position inconnue à l'aide d'un smartphone, requêter la base de données afin de comparer la nouvelle observation RSS et de déduire la position associée aux nouvelles valeurs RSS observées.

#### 3.1.2 Fingerprint hiérarchisé

Nous proposons une solution de fingerprint hiérarchisé, permettant dans un premier temps de déterminer l'étage dans lequel se trouve l'utilisateur, puis de définir la zone plus précise au sein de l'étage identifié et finalement de connaître une position en fonction de l'étage et de la zone déterminée. De plus, un modèle permettant d'identifier l'ascenseur est implémenté, permettant d'assurer la continuité de la localisation à travers plusieurs étages. Ainsi, en divisant le problème en plusieurs modules, nous cherchons à comparer les précisions de localisation par rapport aux solutions de fingerprint en une seule étape.

L'algorithme de localisation hiérarchisé se déroule comme suit :

1. Mesurer les valeurs RSS provenant des 25 balises.
2. Détecter si l'utilisateur se trouve dans l'ascenseur

3. Classifier l'étage avec modèle choisi
4. Classifier la zone en fonction de l'étage avec modèle choisi
5. Classifier la position en fonction de la zone avec modèle choisi

Nous comparerons par la suite l'efficacité de cette architecture en utilisant comme modèles l'algorithme des K plus proches voisins et le perceptron multicouche. Pour ce faire, 21 modèles ont été entraînés pour chaque algorithme. 1 modèle de détection d'ascenseur, 1 modèle de détection d'étage, 6 modèles de détection de zone et 14 modèles de classification de position.

## 3.2 Algorithmes de machine learning

### 3.2.1 K plus proches voisins

L'algorithme des K plus proches voisins (KNN) permet de comparer les mesures RSS récupérées à une position inconnue pendant la navigation aux différentes observations de la base de données de positions connues et attribuer une catégorie à la nouvelle position.

Nous nous concentrons ici sur l'utilisation du K plus proches voisins pondéré. Le déroulement de ce dernier est le suivant :

1. Définir la valeur de K, le nombre de points voisins
2. Calculer la distance définie en (1) entre toute observation  $\mathbf{p}$  du jeu de données et le point de requête  $\mathbf{q} = [rss^1, rss^2, \dots, rss^n]$

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (1)$$

3. Sélectionner les K points d'entraînement les plus proches au point requête
4. Attribuer à chaque point sélectionné un poids  $w_i$  en fonction de sa distance, avec  $w_i = \frac{1}{d(\mathbf{p}, \mathbf{q})}$
5. Déterminer la catégorie  $\hat{y}$  du point requête

$$\hat{y} = \arg \max_c \sum_{i=1}^k w_i \times I(c, y_i) \quad (2)$$

Où  $I(c, y)$  est la fonction d'indication pour la catégorie  $c$

$$I(c, y) = \begin{cases} 1 & \text{si } y \text{ est de catégorie } c \\ 0 & \text{sinon} \end{cases} \quad (3)$$

En calculant la distance euclidienne entre un point d'observation de données RSS et toutes les observations du jeu de données, il est possible de déterminer les K observations les plus proches et ainsi, attribuer la catégorie majoritaire des K observations à la nouvelle mesure. En utilisant une pondération en fonction des distances, les points ayant une faible distance sont favorisés et ceci permet de définir la catégorie du point à classer de manière plus fiable lorsque les distances des K points sont variées.

### 3.2.2 Perceptron multicouche

Les réseaux de neurones artificiels sont des bons candidats pour la tâche de fingerprint car leur capacité d'approximer des

relations non linéaires est particulièrement utile dans le cas de données RSS complexes.

Nous étudierons une architecture spécifique de réseau de neurones : le perceptron multicouche (MLP). Cette architecture est composée de plusieurs unités de calcul, les neurones artificiels. Ce sont des fonctions mathématiques permettant d'établir une fonction de prédiction linéaire en prenant un vecteur  $\mathbf{x}$  en entrée. La somme pondérée des éléments d'entrée est calculée et cette somme est ensuite utilisée comme paramètre d'entrée d'une fonction d'activation non-linéaire  $\phi$ .

$$a = \phi\left(\sum_{i=1}^n w_i x_i + b\right) \quad (4)$$

où :

- $a$  est la valeur obtenue en sortie d'un neurone,
- $n$  est le nombre de valeurs d'entrée du neurone,
- $w_i$  est le poids correspondant à la valeur  $x_i$  reçue en entrée du neurone,
- $b$  est le biais du neurone

Le perceptron multicouche combine ainsi plusieurs neurones en trois couches successives afin d'apprendre des relations complexes entre les valeurs d'entrée et les catégories à prédire. L'architecture étudiée définissant le modèle de fingerprint est la suivante : la couche d'entrée est composée de 25 neurones, un neurone pour chaque valeur RSS observée. La couche cachée est composée de 100 neurones. Ces deux premières couches utilisent la fonction d'activation  $f(x) = \max(0, x)$ . La dernière couche est composée d'autant de neurones que de catégories (6 pour la classification des étages, 1 à 4 pour la classification des zones en fonction de l'étage et 1 à 10 pour la classification de position repère en fonction de la zone). Les neurones de la couche de sortie utilisent la fonction d'activation définie en (5) et permettent ainsi d'obtenir la probabilité associée à chaque catégorie.

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^M e^{x_j}} \quad (5)$$

Les perceptrons multicouches ont comme objectif de minimiser l'erreur de prédiction définie par une fonction de coût : l'entropie croisée  $H(p, q) = - \sum_{x \in \text{classe}} p(x) \log q(x)$ . Les poids des neurones sont donc modifiés par descente de gradient afin de minimiser l'erreur globale. Ainsi, en apprenant les meilleurs poids des neurones le perceptron multicouche est capable de déterminer la position d'un utilisateur en fonction des valeurs RSS.

## 4 Résultats et discussion

Dans cette section, nous présentons les résultats obtenus par les différentes architectures proposées sur de nouvelles données. La tableau 1 résume les résultats obtenus sur les 10 trajets test effectués. L'erreur moyenne pour un trajet est obtenue en calculant la moyenne de la distance haversine entre la position prédite  $\hat{y}$  et la position réelle  $y$  au pas de temps  $t$  définie par :

$$\frac{\sum_{t=1}^T d_{hav}(\hat{y}, y)}{T} \quad (6)$$

Où la valeur de  $T$  varie en fonction du trajet et  $d_{hav}$  entre deux points est calculé à l'aide de la fonction haversine inverse définie comme suit :

$$d = 2r \arcsin \sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos \varphi_1 \cos \varphi_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \quad (7)$$

avec  $\varphi_1$  et  $\varphi_2$  les latitudes des deux points et  $\lambda_1$  et  $\lambda_2$  les longitudes des deux points et  $r$  le rayon de la Terre.

TABLE 1 – Précision des modèles du fingerprint hiérarchisé

Trajet	Précision d'étage (%)		Précision de zone (%)		Précision de position (%)	
	KNN	MLP	KNN	MLP	KNN	MLP
1	100	100	98	91	96	82
2	100	100	97	80	43	20
3	100	100	100	96	73	12
4	100	100	100	70	88	25
5	100	100	100	90	87	46
6	100	100	100	88	61	23
7	100	100	97	82	56	43
8	100	100	100	95	86	56
9	100	100	100	100	59	50
10	100	100	100	90	57	14

De très bonnes performances de classification d'étage et de zone sont obtenues avec les deux algorithmes utilisés. Malgré des faibles précisions de classification des positions, les erreurs moyennes observées sont inférieures à 2 mètres sur les différents trajets test dans le cas du fingerprint hiérarchisé par K plus proches voisins. Ces erreurs de classification sont dans 71% des prédictions des erreurs de classification avec une position voisine.

Nous observons une erreur moyenne plus importante dans le cas du trajet 2 due à une classification de zone erronée sur plusieurs valeurs, impliquant une prédiction de position dans la zone erronée, avec 3.28 m entre la position prédite et la position réelle.

TABLE 2 – Moyenne et (écart type) de l'erreur (m) des solutions de fingerprint

Trajet	Fingerprint hiérarchisé		Fingerprint simple			
	KNN	MLP	KNN		MLP	
1	0.82 (0.96)	0.47 (0.93)	0.31 (0.77)	0.28 (0.75)		
2	1.84 (1.13)	2.01 (1.16)	2.07 (1.71)	1.28 (1.28)		
3	1.34 (0.77)	1.44 (0.64)	1.60 (0.91)	1.33 (0.69)		
4	0.97 (1.10)	1.55 (1.01)	1.05 (1.02)	0.70 (0.84)		
5	0.69 (0.86)	1.13 (1.12)	0.89 (0.95)	0.82 (1.06)		
6	1.35 (1.09)	1.72 (1.01)	1.87 (1.16)	1.63 (1.04)		
7	1.30 (1.62)	1.34 (1.41)	1.76 (1.69)	1.54 (1.59)		
8	0.81 (0.96)	0.95 (1.17)	1.08 (0.98)	1.03 (1.17)		
9	1.01 (1.35)	1.19 (1.35)	1.18 (1.33)	0.97 (1.09)		
10	0.94 (0.78)	1.40 (0.68)	1.48 (1.43)	1.55 (1.03)		
Resultat global :	1.18 (1.01)	1.24 (1.11)	1.25 (1.27)	1.04 (1.09)		

Le fingerprint hiérarchisé utilisant les modèles de perceptron multicouche présente un taux d'erreur de classification plus important au niveau des zones, ce qui détériore en conséquence les résultats obtenus par la classification de position.

De plus, le tableau 2 récapitulatif des erreurs obtenus par les modèles souligne de meilleurs résultats de localisation obtenus par le perceptron multicouche simple. Ce dernier présente également l'avantage de la faible latence pendant la phase de localisation. Même si nous n'avons pas des résultats quanti-

tatifs, il apparaît clairement que la procédure de comparaison avec toute la base de référence du K plus proches voisins prend plus de temps que l'application de la fonction entraînée par le perceptron multicouche dans le cas d'une base de taille conséquente.

## 5 Conclusion

Nous avons présenté dans un article les performances expérimentales de localisation dans un environnement intérieur utilisant des mesures de RSS de la technologie BLE. La méthode d'empreinte radio a été implémentée avec des algorithmes d'apprentissage automatique en deux architectures. Une architecture de fingerprint simple dans laquelle la position est déduite directement des mesures de puissances et une architecture hiérarchisée dans laquelle la localisation se fait avec des niveaux croissants de précision. Les résultats expérimentaux montrent une précision moyenne inférieure ou égale à 2 mètres. Les algorithmes d'apprentissage testés dans cette étude permettent d'atteindre des performances similaires en termes de précision de localisation, mais le perceptron multicouche possède un avantage certain en terme de latence. Enfin, l'architecture de localisation hiérarchisée permet d'atteindre des niveaux de localisation flexibles et adaptables à différents types d'applications.

## Références

- [1] A. Küpper. *Location-Based Services : Fundamentals and Operation*. John Wiley & Sons, 2005.
- [2] J. Schiller et A. Voisard *Location-Based Services*. Elsevier, 2004.
- [3] B. Rao and L. Minakakis *Evolution of Mobile Location-Based Services*. Communications of the ACM, vol. 46, no. 12, pp. 61–65, 2003.
- [4] N. Tadayon, M. T. Rahman, S. Han, S. Valaee, et W. Yu *Decimeter Ranging with Channel State Information*. IEEE Transactions on Wireless Communications, 2019.
- [5] M. Kotaru, K. Joshi, D. Bharadia, et S. Katti *Decimeter Level Localization using WiFi*. ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp.269–282, 2015.
- [6] M. Brunato and R. Battiti *Statistical learning theory for location fingerprinting in wireless lans*. Computer Networks, vol. 47, no. 6, pp. 825–845, 2005.
- [7] X. Liang, X. Gou, and Y. Liu *Fingerprint-based location positioning using improved knn*. 3rd IEEE international conference on network infrastructure and digital content, 2012.
- [8] W. Njima, I. Ahriz, R. Zayani, M. Terré, R. Bouallegue *Deep CNN for Indoor Localization in IoT-Sensor Systems*. Sensors 19(14) : 3127, 2019.
- [9] I. Ahriz, M. Terré, W. Njima *Autoencoder Matrix Completion Based Indoor Localization*. ACSSC 2020 : 1323-1326, 2020.