Titre: Title:	Exploitation des protocoles WiFi pour le décompte automatique des personnes aux alentours d'un point donné
Auteur: Author:	Erwan Nisol
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Nisol, E. (2022). Exploitation des protocoles WiFi pour le décompte automatique des personnes aux alentours d'un point donné [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10555/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/10555/
Directeurs de recherche: Advisors:	Richard Labib
Programme: Program:	Maîtrise recherche en mathématiques appliquées

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Exploitation des protocoles WiFi pour le décompte automatique des personnes aux alentours d'un point donné

ERWAN NISOL
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques appliquées

Août 2022

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Exploitation des protocoles WiFi pour le décompte automatique des personnes aux alentours d'un point donné

présenté par **Erwan NISOL**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Luc ADJENGUE, président

Richard LABIB, membre et directeur de recherche

Antoine SAUCIER Antoine, membre

DÉDICACE

À tous mes amis du labos,
vous me manquerez...

REMERCIEMENTS

Je tiens à exprimer ma reconnaissance à toutes les personnes m'ayant aidé lors de ce projet.

Je voudrais, avant toute chose, remercier mon directeur de recherche, Monsieur le Professeur Richard Labib, pour son soutien précieux et indéfectible du point de vue académique, rédactionnel et personnel.

Je souhaite également remercier toute l'équipe pédagogique de Polytechnique Montréal et de l'X, responsables de ma formation.

Je remercie également l'équipe de l'entreprise Buspas pour m'avoir donné l'opportunité de travailler dans leur bureaux dans le cadre d'un stage, et ce malgré le contexte pandémique.

Je remercie aussi MITACS pour leur soutien tout au long de ce projet.

Enfin, j'aimerais exprimer mes plus sincères remerciements aux personnes suivantes :

- Monsieur Yacin Belmihoub Martel pour son soutien en matière de programmation et de connaissances académiques.
- Monsieur Alexandre Dorais pour les mêmes raisons.
- Madame Hannah Wood pour son soutien rédactionnel en anglais.
- Ma famille et toute personne ayant accepté de relire ce document.

RÉSUMÉ

La ville moderne est sans cesse confrontée à de nombreux défis. Parmi eux figurent notamment son accessibilité en transports et la satisfaction de ses usagers rendues de plus en plus difficile par un nombre grandissant de contraintes. En effet, en plus des limites financières, il est aujourd'hui impossible d'ignorer l'impact environnemental des transports en commun dans la situation actuelle de crise climatique. Une meilleure compréhension du comportement des utilisateurs des autobus est critique pour permettre plus d'optimisations du réseau de transports. Par exemple, la connaissance en temps réel du nombre de personnes patientant à chaque arrêt de bus pourrait mener à l'émergence de transports en commun plus flexibles et moins coûteux. Principalement motivés par cette problématique, nous proposons une méthode de comptage d'appareils connectés présents autour d'un point donné exploitant les protocoles de communication WiFi. Antérieurement à 2013, les appareils connectés communiquaient systématiquement un identifiant unique appelé adresse MAC lorsqu'ils sondaient leurs alentours à la recherche de fournisseurs d'accès, dans des paquets de données appelés "Requêtes de Sonde". Cela rendait alors le décompte de ces appareils aisé. Cependant, depuis 2013, ce décompte a été rendu difficile par l'introduction de l'utilisation quasi-systématique d'une fausse adresse MAC aléatoire. Dans ce mémoire, nous avons mis au point une approche fondée sur l'apprentissage d'une métrique adaptée à l'espace des Requêtes de Sonde. Cette métrique est ensuite exploitée dans un algorithme de partitionnement. Un estimateur du maximum de vraisemblance est ensuite utilisé pour prédire le nombre d'appareils et établir un intervalle de confiance. Les résultats expérimentaux menés sur les données WiFi publiques de CRAWDAD montrent que notre méthode atteint une erreur relative de 21%, et ce sans accéder à l'adresse MAC.

ABSTRACT

In cities around the world, the public transportation sector plays a pivotal role in the daily lives of many. Understanding the behaviour of a city's ridership in real-time is essential to reduce its operating costs and enhance passenger satisfaction. Bus systems, specifically, are notorious for extended waiting times, overcrowding and ineffective routes. Data collected based on passenger behaviour allows for greater flexibility in bus scheduling and helps reduce the burden of single passengers on the overall transport network. A solution for assessing travel demand in real-time is to count the number of people waiting at each bus stop to forecast future load on the system. Mainly motivated by these considerations, this thesis proposes an approach using WiFi communication protocols to estimate the number of connected devices present around a given point. In the past, WiFi-based methods were made easier by the existence of a unique identifier called "Media Access Control" (MAC) address which was systematically transmitted by connected devices to discover access providers surrounding them. Since 2013, the use of a randomized MAC address has become more widespread making tracking and counting devices much harder. Fortunately, previous work from Vanhoef et. al in 2016 has shown that this randomization process is not enough to prevent tracking. In their article, the authors introduced a clustering algorithm they claim capable of tracking perfectly 50% of devices over a span of 20 minutes. In 2020, another clustering algorithm called DBSCAN used by Uras et al. was able to count the number of devices with an accuracy of 65.2%. In this thesis, a metric learning and Bayesian modeling is proposed to improve the existing literature. A maximum likelihood estimator is built on the calculated clusters and assigned a confidence interval. Experimental results on the public WiFi dataset, CRAWDAD, has shown that the proposed method achieves a relative error of 21% using each clustering algorithm, without access to the MAC address.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

RS	Requête de sonde (Probe request en anglais)
MAC	Media Access Control
SSID	Service Set Identifier
RSSI	Received Signal Strength Indicator
SN	Sequence Number
CRAWDAD	Community Resource for Archiving Wireless Data
DBSCAN	Density-based spatial clustering of applications with noise
CSI	Channel State Information
EMV	Estimateur du Maximum de Vraisemblance
EAMP	Erreur Absolue Moyenne en Pourcentage
IEEE	Institute of Electrical and Electronics Engineers
OUI	Organizationally Unique Identifier
U/L bit	Universal/Local bit

CHAPITRE 1 INTRODUCTION

Selon le rapport de gestion du Ministère des Transports (2020-2021), au Québec, plus de 275 millions de déplacements en transport en commun sont effectués chaque année. Pour beaucoup de citoyens, ces trajets représentent une part significative de la vie quotidienne. L'industrie du transport collectif impacte donc profondément les usagers et les villes, à la fois sur le plan financier, environnemental, mais aussi sur le plan du bien-être.

Cela est d'autant plus vrai que, selon l'enquête sociale générale de Statistique Canada (2010), les usagers de transports en commun sont bien plus susceptibles que les automobilistes de se dire insatisfaits de leur temps de déplacement pour se rendre sur leur lieu de travail (23% des usagers des transports contre 18% des automobilistes).

L'optimisation des réseaux de transport pour réduire la durée des trajets ainsi que les temps d'attente devient alors un enjeu de premier plan pour la ville moderne. Or, de telles optimisations ne sont possibles qu'avec une meilleure connaissance de la demande des usagers des transports collectifs. Par exemple, il serait très intéressant pour les opérateurs de transports en commun de connaître cette demande au travers de la ville, et ce en tout instant.

Principalement motivés par cette problématique, nous présentons donc dans ce mémoire une approche pour réaliser automatiquement et en temps réel le décompte des personnes présentes aux alentours d'un point donné.

1.1 Objectif et enjeux

Notre objectif est de créer une méthode qui, lorsque mise en oeuvre dans un système embarqué indépendant en énergie, permet d'estimer le nombre de personnes présentes aux alentours d'un point donné en temps réel.

À cet objectif est tout d'abord associé un enjeu social. Appliquée aux arrêts de bus, la connaissance en temps réel de la demande des passagers ouvre la voie à une flexibilisation des routes empruntées, permettant une meilleure desserte de la zone couverte par les transports et donc une meilleure satisfaction des usagers.

Il en découle également un enjeu financier. En effet, toujours dans le contexte de l'acheminement de personnes, l'optimisation des réseaux de transport peut permettre de fournir le même service aux citoyens mais avec une flotte d'autobus réduite. Cela permet de réduire les coûts opérationnels des réseaux et de potentiellement diminuer le prix des transports pour les usagers.

De surcroît, un enjeu environnemental se dégage. En effet, une taille de flotte d'autobus réduite signifie également un impact réduit sur la ville en termes de pollution atmosphérique et de nuisance sonore.

Enfin, dans un contexte pandémique, un enjeu sanitaire apparaît : l'estimation du nombre de personnes peut contribuer au respect des limites d'accueil de certains lieux et aider à l'endiguement des propagations virales.

De nombreux travaux ont été menés pour répondre à des problèmes similaires. Des approches très différentes ont été mises au point afin de réaliser un décompte de personnes. Nous présentons ici les méthodes principales, leurs avantages et inconvénients.

1.2 Forces et faiblesses des méthodes existantes

Les méthodes les plus directes et efficaces sont visuelles. Elles font appel à des caméras de surveillance dont les images sont exploitées par un réseau de neurones convolutif pour en extraire le décompte des personnes. Bien qu'elles soient de loin les plus précises et fiables, dépassant des précisions de 75% (Raghavachari et al. (2015)), ces approches présentent des faiblesses qui les rendent difficilement exploitables dans notre cas d'étude. En effet, l'utilisation de caméras ainsi que la puissance de calcul nécessaire pour traiter les images en temps réel rendent cette approche particulièrement énergivore. Cela n'est pas compatible avec la nécessité d'une solution autonome en énergie. À cela s'ajoutent des problématiques de respect de la vie privée et de droit à l'image.

Il existe également des méthodes utilisant les perturbations des ondes WiFi entre un émetteur et un récepteur. L'atténuation et le déphasage mesurés sur chaque fréquence porteuse du signal (appelé CSI) permet ensuite, via des méthodes d'apprentissage machine, de déduire le nombre de corps aux alentours (Oshiga et al. (2019) ou encore Choi et al. (2021)). Cependant, ce type de méthode n'est pas exploitable dans le cas d'un système embarqué en raison de la nécessité d'avoir un émetteur et un récepteur spatialement séparés.

Enfin, il existe des méthodes passives indirectes exploitant les protocoles de connexion Bluetooth ou WiFi pour estimer le nombre d'appareils connectés présents aux alentours. En effet, les téléphones dont le WiFi est activé envoient en permanence des paquets de données appelés "Requête de Sonde" (RS) pour rechercher des fournisseurs d'accès internet. Collecter ces paquets se fait passivement et requiert très peu d'énergie. Exploiter les RS était très simple et particulièrement fiable avant 2013 (Vanhoef et al. (2016)). Cependant, depuis, les fabricants d'appareils connectés ont mis en place des mécanismes pour rendre plus difficile l'espionnage des téléphones par un tiers. En conséquence, ces méthodes sont aujourd'hui très peu précises

pour le décompte d'appareils connectés, en particulier lorsqu'ils sont peu nombreux (65.2% de précision atteints par Uras et al. (2020) avec DBSCAN). De plus, ces méthodes reposent sur l'hypothèse qu'à chaque personne est associée un unique téléphone.

Étant données les approches listées précédemment ainsi que leurs forces et leurs faiblesses, et au regard des contraintes impliquées par notre problématique, les seules méthodes exploitables dans notre cas d'étude sont celles utilisant les protocoles de connexion WiFI.

1.3 Méthode envisagée et sous-objectifs associés

Nous souhaitons donc améliorer les méthodes utilisant les RS pour le décompte des appareils connectés, et les rendre plus rapides et plus adaptées pour notre cas d'étude.

Il s'agit d'un problème difficile, car même si les connexions WiFi entre les appareils sont régies par un protocole défini, les téléphones ont aujourd'hui tendance à envoyer un nombre réduit d'informations lorsqu'ils sont à la recherche de fournisseurs d'accès internet. Certains vont même jusqu'à rendre ces informations aléatoires afin de perturber un éventuel traqueur.

Nous proposons donc trois sous-objectifs pour la mise en place de notre méthode.

Tout d'abord, pour regrouper les RS par téléphone d'origine, il est nécessaire de disposer d'une métrique permettant de relier le contenu de deux RS différentes à la possibilité qu'elles proviennent du même appareil. Cette métrique doit être robuste et être peu influencée par les mécanismes de randomisation du contenu de certaines RS. Idéalement, elle doit nécessiter pas ou peu d'ajustements pour pouvoir tenir compte des changements futurs des protocoles de communication sans fil.

Objectif 1 :
L'élaboration d'une nouvelle métrique permettant de comparer deux Requêtes de Sonde (RS) et donc de mesurer leur similarité.

Ensuite, une fois la similarité deux à deux des RS évaluée, il faut choisir une méthode pour leur regroupement. Les algorithmes de partitionnement existants tels que DBSCAN, OPTICS ou K-means, fonctionnent généralement bien quel que soit le type de données. Nous sélectionnerons les algorithmes et les modifierons pour améliorer leurs performances. Ces performances seront évaluées par deux critères :

— Complexité réduite, afin qu'ils soient utilisables en temps réel.
— Précision.

Objectif 2 :
L'amélioration de méthodes existantes pour le regroupement des RS par téléphone d'origine.

Enfin, une fois les données regroupées, l'évaluation de l'erreur permettra de mettre en place un estimateur ainsi que des intervalles de confiance sur le nombre de téléphones détectés.

Objectif 3 :
L'analyse de l'erreur commise sur le nombre d'appareils détectés.

La Figure 1.1 présente notre approche générale et l'articulation des objectifs. Tout d'abord, il faudra obtenir des données de Requêtes de Sonde. Une fois décodées, ces données seront analysées afin d'élaborer une métrique permettant de comparer les RS deux à deux. Ensuite, plusieurs algorithmes de partitionnement seront comparés. Enfin, nous analyserons l'erreur afin d'établir un intervalle de confiance sur la prédiction du nombre de téléphones.

La Figure 1.2 présente le fonctionnement général de l'approche proposée. Un nombre inconnu d'appareils émettent des Requêtes de Sonde. Celles-ci sont collectées et décodées. Ensuite, la métrique mise au point dans le cadre de l'objectif 1 est utilisée afin de calculer les distances deux à deux entre les RS. Ces distances sont alors utilisées par l'algorithme de partitionnement retenu pour l'objectif 2. Le résultat du partitionnement est ensuite utilisé pour donner un intervalle de confiance sur le nombre d'appareils détectés.

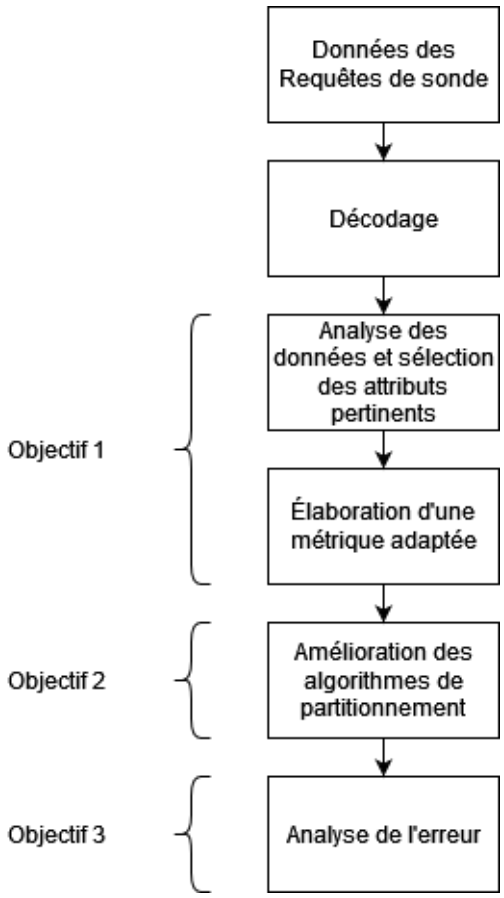

Figure 1.1 Objectifs et approche générale pour les réaliser

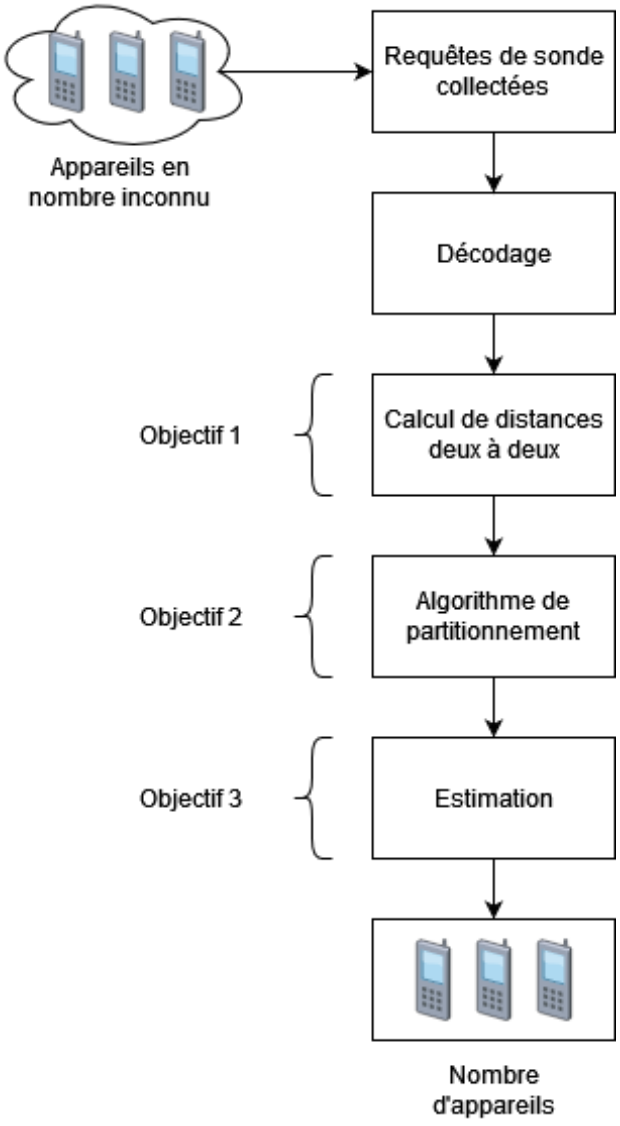

Figure 1.2 Vue d'ensemble de l'approche proposée

CHAPITRE 2 REVUE DE LITTÉRATURE

Il existe de très nombreuses méthodes de comptage de personnes dans l'état de l'art, reposant sur une variété de principes. On peut les diviser en deux grandes catégories selon le type de capteur qu'elles utilisent :

— Les méthodes visuelles, utilisant des caméras visibles ou infra-rouge. Ce sont les méthodes les plus répandues car elles constituent une approche naturelle au comptage de personnes. Cependant, elles reposent souvent sur des algorithmes d'apprentissage profond nécessitant beaucoup de puissance de calcul pour fonctionner en temps réel. Cela entraîne des répercussions sur la consommation énergétique de ces solutions. De plus, les méthodes visuelles sont limitées par le champ de vision des capteurs et les occlusions. Dans cette revue de littérature, nous présentons quelques-unes de ces méthodes mais n'en ferons pas usage pour des raisons de contraintes énergétiques dans nos cas d'étude.

— Les méthodes radio, utilisant une carte WiFi ou Bluetooth pour analyser les ondes aux alentours et tenter d'en déduire le compte de personnes. Ces méthodes requièrent souvent peu d'énergie pour fonctionner, mais la nature indirecte du comptage effectué les rend souvent moins précises. Ce sont principalement ces méthodes que nous présentons dans ce chapitre, car elles sont souvent plus adaptées à nos cas d'étude pour des raisons énergétiques.

Nous allons maintenant présenter les méthodes de l'état de l'art au meilleur de nos connaissances. Nous commençons par les méthodes visuelles, avant de nous pencher sur les méthodes reposant sur les ondes.

2.1 Méthodes visuelles

L'approche naturelle pour le comptage automatique des personnes est d'exploiter des images de vidéo-surveillance. Dans cette section, nous énumérons quelques algorithmes très utilisés ainsi que des jeux de données adaptés à un contexte de comptage de personnes.

De nombreux algorithmes utilisent des réseaux de neurones profonds avec de multiples couches de convolution. Ils permettent de détecter et localiser des objets sur des images, et ce en temps réel. Un exemple classique d'un tel algorithme est "You Only Look Once" proposé par Redmon et al. (2016) qui peut facilement traiter plusieurs images voire dizaines d'images par seconde. Ces algorithmes sont en constante évolution et bénéficient de larges jeux de données annotées comme COCO (Lin et al. (2014)), Eurocity Persons (Braun et al.

(2019)) ou encore CrowdHuman (Shao et al. (2018)) pour s'entraîner. Certains modèles plus récents comme CrowdDet (Chu et al. (2020)) parviennent à atteindre des précisions moyennes dépassant les 90% même dans des contextes difficiles avec des foules denses d'êtres humains.

D'autres méthodes spécifiquement développées pour la détection de personnes utilisent des capteurs infrarouges basse résolution afin d'isoler plus facilement les corps chauds. Des techniques d'apprentissage profond sont ensuite utilisées pour estimer le compte. Une telle méthode est présentée dans "An infrared array sensor-based method for localizing and counting people for health care and monitoring" (Bouazizi and Ohtsuki (2020)) par exemple, avec une précision de 97% pour les comptes de zéro, une, deux ou trois personnes. Une application proposée de cette méthode est la surveillance de personnes âgées.

Les images infrarouges peuvent également être combinées à des images visibles pour obtenir un compte encore plus précis, même dans le cas de foules très denses. Une telle approche est proposée dans "People counting using visible and infrared images" (Filipic et al. (2021)) par exemple, où l'image infrarouge est simplement traitée comme le 4ème canal d'une image RGB.

Cependant, toutes les méthodes fondées sur la visualisation requièrent des caméras, ce qui introduit plusieurs difficultés :

— La gestion de l'énergie. Les caméras et le traitement de leurs images demandent beaucoup de puissance. Cette énergie n'est pas toujours facilement disponible dans le contexte d'un système embarqué.
— La difficulté de couvrir intégralement les zones d'intérêt avec une seule caméra tout en diminuant les risques d'occlusion.
— Le risque d'atteinte à la vie privée des personnes filmées.

Pour toutes ces raisons, nous ne ferons pas usage des méthodes visuelles et n'élaborerons pas plus sur ce sujet. Les méthodes les plus adaptées à nos cas d'étude sont celles utilisant les ondes, et c'est sur celles-ci que nous allons nous concentrer désormais.

2.2 Méthodes utilisant les ondes

Nous allons maintenant présenter les techniques de comptage de personnes reposant sur les ondes radio. Ces approches peuvent se subdiviser en quatre catégories :

— Les méthodes actives, qui exploitent les protocoles de communication WiFi. Elles permettent d'attaquer les appareils à proximité pour les pousser à révéler leur présence.
— Les méthodes passives exploitant les protocoles de communication WiFi pour compter le nombre d'appareils aux alentours. Ce compte peut ensuite être extrapolé à une

estimation du nombre de personnes.
— Les méthodes passives exploitant les protocoles de communication Bluetooth de manière similaire.
— Les méthodes passives exploitant le "Channel State Information" (CSI). Cette information est une mesure de la transformation des ondes radio entre un émetteur et un récepteur. Elle change donc en fonction de l'environnement, et des modèles peuvent être utilisés pour détecter des personnes.

Nous parlerons dans un premier temps des dispositifs de capture de données radio, puis nous présenterons plusieurs articles pertinents pour chacune des catégories énoncées ci-dessus.

2.2.1 Dispositifs d'acquisition

L'application de la plupart des méthodes que nous présenterons dans les parties suivantes requiert des dispositifs d'acquisition. Deux types de données nous intéressent.

Les premières données exploitées sont des paquets de données WiFi. Pour les intercepter, une approche possible est d'utiliser un système muni d'une carte WiFi qui surveille et décode les paquets de données émis aux alentours. L'article "A Case Study of WiFi Sniffing Performance Evaluation" (Li et al. (2020)) propose et compare des structures et des réglages pour un tel système d'acquisition. L'article propose notamment une stratégie de "saut de canal" permettant de surveiller efficacement tous les canaux de transmission des signaux WiFi. Ce même système permet également de créer des paquets et de les émettre.

Le second type de données utilisées dans certains des articles que nous présenterons est le "Channel State Information". Le CSI est une information sur le déphasage et le changement d'amplitude de chacune des ondes sous-porteuses du signal. Un exemple d'outil permettant son extraction est rendu disponible par Halperin et al. (2011).

Ces dispositifs sont utilisés dans la plupart des articles que nous citons dans cet état de l'art.

2.2.2 Méthodes actives

Les premières méthodes que nous présentons sont celles dites actives. Elles sont fondées sur l'émission de paquets et l'analyse du trafic résultant. Souvent, elles exploitent des failles d'implémentation des protocoles de communication WiFi et sont qualifiées "d'attaques".

Un premier exemple de méthode active est l'attaque Karma, proposée dans “Attacking automatic wireless network selection” (Dai Zovi and Macaulay (2005)). Cette attaque consiste à faire croire aux appareils alentours qu'un fournisseur d'accès internet avec un nom courant est

présent (par exemple "FreeWifi", "Linksys", "NETGEAR" qui sont parmi les plus communs au Canada). Tous les appareils configurés pour se connecter automatiquement à un réseau portant l'un de ces noms vont alors entamer un processus d'authentification en utilisant leur adresse MAC unique, ce qui permet de les compter. Cette méthode requiert une base de données des noms de réseaux WiFi les plus communs dans une zone donnée. Une expérience précédemment menée par Vanhoef et. al. dans "Why MAC Address Randomization is Not Enough : An Analysis of WiFi Network Discovery Mechanisms" (Vanhoef et al. (2016)) a réussi à obtenir une réaction de 17.4% des appareils en utilisant cette attaque et une liste des noms de réseaux les plus populaires au Canada. Cela ouvre la voie à un comptage actif utilisant cette méthode pour détecter les appareils puis faisant une estimation du compte de personnes en utilisant un facteur d'extrapolation. Cependant le faible nombre d'appareils réactifs ne permettra pas d'avoir une précision suffisante dans les cas où peu de téléphones (c'est à dire moins d'une dizaine) sont présents aux alentours. Or c'est un ordre de grandeur tout à fait possible pour certains cas d'étude.

Une autre méthode active est appelée "inversion de l'UUID". Certains appareils transmettent des informations WPS ("WiFi Protection Setup") lorsqu'ils sondent les alentours à la recherche de fournisseurs d'accès internet. Ces informations ont pour but de faciliter l'appairage avec une station. Or un des champs contenus par les informations WPS est un identifiant appelé "Universally Unique Identifier" (UUID). Ce champ est calculé de manière déterministe à partir de l'adresse MAC unique de l'appareil. Il peut donc être inversé pour retrouver l'adresse MAC en utilisant de grandes tables de données pré-calculées. Cependant, la transmission des informations WPS est de plus en plus rare avec les nouveaux téléphones et systèmes d'exploitation. Notamment, l'UUID n'est plus du tout transmis depuis la version 8 d'Android selon Fenske et al. (2021). Cette méthode n'est donc plus exploitable pour le comptage des appareils récents et futurs.

Il existe enfin une méthode appelée "RTS/CTS" qui exploite l'envoi de paquets appelés "Ready To Send" contenant une adresse MAC spécifiquement visée. L'adresse MAC est un identifiant unique de chaque appareil. Si l'appareil avec l'adresse MAC correspondante est à proximité, il répondra par un paquet appelé "Clear To Send" trahissant alors sa présence. Les auteurs de "A study of mac address randomization in mobile devices and when it fails" (Martin et al. (2017)) montrent qu'en 2017, 100% des appareils testés étaient vulnérables à cette attaque, et ce parfois même si leur WiFi était désactivé. Les travaux de Fenske et al. (2021) montrent que cette faille majeure est encore observée dans quelques appareils, notamment certains modèles de téléphones des marques LG et Motorola plus récents que 2018. Néanmoins, la majorité des appareils récents ne sont plus vulnérables à cette attaque, ne la rendant pas exploitable dans le cas général. De plus, cette attaque ne fonctionne que si l'adresse MAC

réelle de la cible est connue, ce qui n'est pas le cas dans nos cas d'étude.

Pour des raisons de considérations éthiques, nous ne ferons usage d'aucune de ces attaques pour nos travaux. Nous allons maintenant présenter les méthodes passives utilisant le WiFi.

2.2.3 Méthodes passives utilisant le WiFi

Il existe des méthodes passives qui reposent uniquement sur l'écoute du trafic WiFi. En effet, les appareils dont le WiFi est activé sondent régulièrement leur environnement pour demander aux fournisseurs d'accès potentiels de s'identifier. Ces messages appelés "Requêtes de Sonde" sont l'objet de plusieurs méthodes de décompte et suivi d'appareils. En effet, très peu de téléphones limitent les informations transmises par les Requêtes de Sonde afin de le rendre plus difficile à distinguer (Fenske et al. (2021)), ce qui peut permettre de les regrouper par appareil émettant et en déduire le nombre de téléphones alentours ou encore leur temps de présence.

Avant 2013, il était très simple de compter ces appareils et de les suivre avec une précision très élevée, car tous les paquets de données émis contenaient un indicateur unique propre à chaque appareil et appelé adresse MAC. Cependant, l'utilisation de fausses adresses MAC aléatoires s'est depuis largement démocratisé comme le montrent Fenske et al. (2021) dans “A Case Study of WiFi Sniffing Performance Evaluation". Dans cet article, les auteurs mènent une étude du comportement d'émission de paquets WiFi sur 160 modèles de téléphones différents. Malgré cette difficulté nouvelle, il existe tout de même de nombreuses approches pour estimer le nombre d'appareils aux environs.

Plusieurs méthodes précédemment développées utilisent une approche de ce type. Par exemple, "Estimating bus passenger volume based on a WiFi scanner survey" (Hidayat et al. (2020)) propose d'utiliser une carte WiFi embarquée dans un autobus afin de compter les passagers à bord. Leur approche combine des données GPS avec les adresses MAC transmises aux alentours afin d'associer une vitesse à chacune de ces adresses. Ces vitesses sont ensuite comparées à celle du bus pour classifier chaque adresse MAC comme provenant soit du bus, soit de l'extérieur. Il en est déduit une estimation du compte de passagers ne prenant pas en compte les piétons et autres sources potentielles de paquets WiFi extérieures au bus. Cette approche n'est pas exactement adaptée à notre cas d'étude, puisque nous cherchons à compter les personnes près d'une station fixe. De plus, elle ne semble pas prendre en compte la randomisation des adresses MAC.

Une autre méthode, celle de "Crowd Forecasting based on WiFi Sensors and LSTM Neural Networks" (Singh et al. (2020)) utilise plusieurs stations fixes, munies de cartes WiFi et

placées à différents endroits afin de compter le nombre de personnes dans une foule et de prédire leur déplacement à l'aide de réseaux de neurones récurrents appelés LSTM. Cette approche exploite une indication de puissance du signal appelée RSSI, mesurée par chaque station lorsqu'un paquet de données est reçu, pour estimer la zone dans laquelle se trouve l'appareil émettant. Le compte dans chaque zone est fait en moyennant le nombre de Requêtes de Sonde reçues sur 5 minutes, et en extrapolant ce compte à un nombre de personnes en utilisant un facteur de correction. Cette approche est robuste et immunisée à la randomisation des adresses MAC si elle est appliquée à des foules nombreuses. En effet, travailler avec un grand nombre d'appareils émettant permet de diminuer l'impact des différences de comportement des appareils utilisant le WiFi via la loi des grands nombres. Cela permet aux auteurs d'atteindre un taux d'erreur de l'ordre de 10%. Dans nos cas d'étude, nous nous attendons à des environnements où beaucoup moins de personnes sont attendues, ce qui ne permet pas d'employer directement cette méthode.

Une méthode très similaire présentée dans "Understanding Crowd Behaviors in a Social Event by Passive WiFi Sensing and Data Mining" (Zhou et al. (2020)) fait également usage de multiples stations WiFi pour prévoir les mouvements d'une large foule. Le nombre d'appareils aux alentours de chaque station est estimé en ne prenant en compte que les Requêtes de Sonde contenant une adresse globale et unique, ce qui est indiqué par un certain bit de l'adresse transmise. Les auteurs filtrent ensuite les adresses qui ont déjà été vues trop de fois sur une trop longue période, considérant qu'elles appartiennent à des appareils statiques comme des caméras de surveillance par exemple. Ces données sont ensuite utilisées pour extraire des motifs temporels et des trajectoires privilégiées au sein de la foule. Encore une fois, la précision de cette approche repose sur la taille de la foule étudiée.

Toujours avec plusieurs stations, "Pairing WiFi and Bluetooth MAC Addresses Through Passive Packet Capture" (Longo et al. (2018)) propose une méthode de triangulation fondée sur l'estimation de la distance via l'indicateur de puissance du signal (RSSI). Plusieurs méthodes sont expérimentées, avec des nombres différents de stations d'écoute WiFi. Cependant, le lien entre RSSI et distance à la station d'écoute est très perturbé par les interférences et les rebonds des signaux sur l'environnement. Aussi, cette approche nécessite un grand nombre de stations (6 environ) pour obtenir une triangulation. Dans cet article, l'objectif principal n'est pas de compter les appareils mais de les localiser. Il est cependant envisageable d'utiliser ces localisations pour tenter de compter les émetteurs. Les auteurs réussissent à atteindre une précision de 70% pour la localisation de 15 appareils répartis dans une salle de 50 m^2. Cependant, la contrainte du nombre de stations d'écoute WiFi est importante dans notre cas d'étude. En effet, un système embarqué autonome ne peut pas contenir un tel nombre d'antennes réparties dans des zones différents.

Enfin, certaines méthodes permettent de prendre en compte le contenu des Requêtes de Sonde et d'exploiter une certaine catégorie de champs appelés "Éléments d'Information" pour les regrouper selon leur téléphone d'origine. Une première approche est celle de Vanhoef et al. (2016). Celle-ci définit la "signature" d'un téléphone comme l'ensemble des valeurs prises par les Éléments d'Information. Les Requêtes de Sonde avec des signatures identiques sont ensuite regroupées au fur et à mesure de leur arrivée. Les résultats de cette méthode dépendent beaucoup du nombre de téléphones présents aux alentours. Vanhoef et. al. montrent qu'en 2016, ils étaient capables de suivre parfaitement entre 30 et 50% des appareils sur une durée de 20 minutes si leur nombre ne dépasse pas une soixantaine. Cependant, cette méthode est très vulnérable aux appareils changeant de signature à chaque émission de Requêtes de Sonde. Telle qu'elle est présentée, cette méthode ne permet donc pas de donner directement un compte fiable. Par ailleurs, l'objectif de cette méthode n'est pas le décompte des personnes mais plutôt le suivi de certaines. L'algorithme de Vanhoef et. al. exploite également un champ appelé "Numéro de Séquence", dont on sait qu'il est presque systématiquement rendu aléatoire depuis 2018 Fenske et al. (2021). Néanmoins, l'approche de cet article est très intéressante pour notre cas d'étude. En effet, elle ne nécessite qu'une station d'écoute WiFi. L'article de Vanhoef et al. (2016) est une des références principales pour nos travaux.

Une autre approche utilisant les éléments d'information est celle de "WiFi Probes sniffing : An Artificial Intelligence based approach for MAC addresses de-randomization" de Uras et al. (2020). L'hypothèse faite par les auteurs est que les Requêtes de Sonde en provenance du même téléphone ont des signatures au moins similaires, si elles ne sont pas identiques. Ainsi, plutôt qu'une égalité stricte entre les signatures des Requêtes de Sonde, ce sont des algorithmes de partitionnement tels que DBSCAN ou OPTICS qui sont utilisés pour regrouper les RS par téléphone d'origine. Les auteurs annoncent obtenir une précision de 65.2% pour le décompte des appareils connectés avec DBSCAN. Cependant, la validation du modèle semble avoir été faite sur un ensemble de données de taille très réduite, voire sur une seule capture comportant une vingtaine d'appareils. Tout comme pour l'article précédent, cette méthode semble particulièrement adaptée à nos cas d'étude. En effet, elle n'utilise qu'une seule station et semble donner directement le compte d'appareils connectés aux alentours. L'article de Uras et al. (2020) est également une des références principales pour nos travaux.

Dans la section suivante, nous allons présenter des méthodes très similaires, reposant cette fois sur le Bluetooth plutôt que le WiFi.

2.3 Méthodes passives exploitant le Bluetooth

Il existe un autre standard de communication sans fil dont les téléphones cellulaires modernes sont presque systématiquement équipés : le Bluetooth. Le Bluetooth a un fonctionnement très similaire au WiFi, mais avec moins de portée. En effet, contrairement au WiFi dont la portée peut atteindre une centaine de mètres à l'extérieur, le Bluetooth a une portée beaucoup plus réduite de quelques mètres à quelques dizaines de mètres (Chilipirea et al. (2018)). Le Bluetooth est également beaucoup plus souvent désactivé sur les téléphones que le WiFi. En effet, les auteurs de "Estimating Crowd Densities and Pedestrian Flows Using WiFi and Bluetooth" (Schauer et al. (2014)) ont trouvé que moins de 2.8% des téléphones étaient détectables en utilisant le Bluetooth en 2020. La précision des approches uniquement fondées sur le Bluetooth semble alors encore une fois dépendre de la loi des grands nombres, et n'être envisageable que dans le cadre de foules très denses et proches du capteur.

Cet avantage du WiFi sur le Bluetooth semble être confirmé par l'article "Estimating Crowd Densities and Pedestrian Flows Using WiFi and Bluetooth" (Schauer et al. (2014)). Ses auteurs comparent une approche fondée sur le Bluetooth à trois approches simples fondées sur le WiFi. Cet article montre que le décompte obtenu avec une méthode exploitant à la fois le RSSI et l'adresse MAC des paquets WiFi obtient une meilleure corrélation moyenne avec le vrai nombre d'appareils (0.57) que leur méthode exploitant le Bluetooth (0.44). Cependant, l'article date de 2014 et la randomisation des adresses MAC était alors encore peu répandue. Il n'est donc pas évident que cette comparaison tienne encore à ce jour.

Une autre approche est de combiner l'information obtenue par les paquets Bluetooth et les paquets WiFi. C'est ce que propose "Pairing WiFi and Bluetooth MAC addresses through passive packet capture" (Longo et al. (2018)). Les auteurs montrent que le RSSI obtenu lors des captures de paquets Bluetooth est très similaire à celui du WiFi, permettant d'utiliser à la fois le WiFi et le Bluetooth pour mettre en oeuvre une méthode de triangulation. Cependant ils montrent également que la relation entre le RSSI et la distance spatiale est complexe, et que de nombreux facteurs entrent en compte. Notamment deux téléphones de marques différentes à une dizaine de mètres d'écart peuvent être captés au même RSSI. De plus cette méthode requiert de multiples stations d'écoute placées à différents endroits, ce qui n'est pas adapté à notre cas d'étude.

La section suivante (et dernière de notre état de l'art) présente des méthodes très différentes de celles que nous venons de présenter, utilisant une grandeur mesurable sur les ondes WiFi appelée CSI.

2.3.1 Méthodes passives utilisant le CSI

Le CSI est une information mesurable qui représente les transformations que subit un signal en se propageant dans le milieu entre un émetteur d'onde et un récepteur. Certaines méthodes existantes exploitent cette information pour faire de la détection d'êtres humains et de l'estimation de leur nombre.

C'est le cas de "Human Detection For Crowd Count Estimation Using CSI of WiFi Signals" (Oshiga et al. (2019)) qui, après pré-traitement du CSI, utilise une technique de classification afin d'estimer si le nombre de personnes présentes est de 0, 1, entre 2 et 3, entre 4 et 6 ou entre 7 et 10. Les auteurs parviennent à une précision de 95% dans leurs prédictions. Un des principaux avantages de cette méthode est qu'elle ne repose pas sur la présence ou non de téléphones, mais directement sur la présence des corps qui perturbent la transmission des ondes. Une personne au téléphone éteint ou sans téléphone pourrait alors tout de même être détectée, ce qui n'est pas le cas des méthodes reposant sur les Requêtes de Sonde. Les méthodes fondées sur le CSI ont également l'avantage d'être totalement non intrusives et éthiques car l'identification d'une personne semble impossible. Cependant, il est probable que cette méthode soit perturbée par la présence d'objets non humains, tels que des voitures, des vélos ou des bus. Un autre défaut de cette méthode est qu'elle nécessite un émetteur, par exemple un router sans fil propageant des signaux WiFi. Elle nécessite donc plus de matériel et est plus énergivore que les autres méthodes vues jusqu'ici. Un router WiFi classique a une puissance de 2 à 20 watts, ce qui n'est pas compatible avec un système embarqué indépendant en énergie.

Les auteurs de "Simultaneous Crowd Counting and Localization by WiFi CSI" (Choi et al. (2021)) proposent d'aller plus loin et de localiser les piétons en plus de les compter, toujours en utilisant le CSI. Encore une fois, la méthode proposée commence par le prétraitement du CSI afin de le lisser et de retirer le bruit. Certaines caractéristiques des courbes de CSI sont ensuite extraites et utilisées dans une forêt aléatoire afin de prédire à la fois le compte des personnes et leur position dans une des quatre zones d'expérimentation définies. Cette méthode parvient à une précision d'environ 80% pour la détection d'entre 0 et 5 personnes. Néanmoins, les expérimentations ont été réalisées dans un environnement contrôlé, en l'occurrence une salle de séminaire. On peut s'attendre à des différences significatives en passant à un environnement ouvert, avec beaucoup plus de perturbations du CSI.

Enfin, l'article "Towards People Counting Using WiFi CSI of Mobile Devices" (Mizutani et al. (2020)) montre également le potentiel du CSI, puisque les auteurs parviennent à utiliser une méthode similaire aux deux précédentes afin de compter le nombre d'occupants d'une pièce. Leur approche se fonde sur une simple régression linéaire sur les phases et les amplitudes

du CSI. Ils parviennent à atteindre une erreur quadratique moyenne de 0.5 environ dans le décompte. Cependant, l'article démontre également la très grande dépendance de la méthode au positionnement des récepteurs WiFi. En effet, décaler les récepteurs de moins d'un mètre peut faire passer l'erreur quadratique moyenne à plus de 6. Il semble donc se dégager que les méthodes utilisant le CSI sont très sensibles aux perturbations de leur environnement.

2.4 Conclusion sur l'état de l'art

De très nombreuses approches sont envisageables pour réaliser le décompte de personnes. Cependant, toutes ne sont pas adaptées aux contraintes de notre cas d'étude. En effet, pour que nos travaux soient applicables dans un système embarqué, il faut prendre en compte les considérations suivantes :

- L'énergie disponible est limitée.
- L'espace utilisable est limité.
- L'environnement peut être intérieur comme extérieur et donc très perturbé.
- Le nombre d'appareils à détecter peut être faible (moins de 10), ou assez grand (plus de 50).

Étant données ces contraintes, nous avons pris la décision d'orienter notre travail vers les méthodes passives utilisant le WiFi. En particulier, celles qui regroupent les Requêtes de Sonde entre elles pour en déduire un compte. Pour ces raisons, les principaux articles dont nos travaux sont inspirés sont ceux de Vanhoef et al. (2016) et de Uras et al. (2020).

Le chapitre suivant présente le protocole 802.11 de connexion WiFi plus en détails, ainsi qu'une analyse de données qui nous aidera à construire notre approche.

CHAPITRE 3 PROTOCOLE 802.11 ET RS

L'établissement de connexions entre les appareils WiFi requiert des normes de communication sans fil. La communication de type "Wireless Local Area Network" (WLAN) est la plus utilisée par les appareils utilisant le WiFi. Elle permet l'établissement de connexion sans fil entre des appareils proches. Les normes qui régissent ce protocole de communication sont définies par les standards de l'IEEE 802.11.

Les téléphones portables utilisent notamment ce protocole afin de se connecter à des fournisseurs d'accès reliés à l'internet. Si le WiFi du téléphone est activé et afin de découvrir les fournisseurs d'accès aux alentours, le téléphone enverra de temps en temps des paquets de données appelés "Requêtes de Sonde" pour provoquer la réponse des points d'accès disponibles. Ces derniers envoient alors une "Réponse de Sonde" contenant les informations nécessaires pour démarrer un éventuel processus d'autentification.

L'objectif de ce chapitre est de présenter le protocole 802.11 ainsi que les données existantes à notre disposition. Nous verrons que les Requêtes de Sonde contiennent de nombreux champs, et nous évaluerons la quantité d'information que chacun de ces champs apporte pour le partitionnement des Requêtes de Sonde par téléphone d'origine. Cette information sera utilisée afin de sélectionner les champs les plus pertinents pour l'élaboration d'une métrique sur l'espace des RS, au chapitre suivant.

3.1 Protocole 802.11 : Requêtes de Sonde

Cette section a pour but de présenter les caractéristiques générales des RS. Une fois décodés, les paquets qui en résultent contiennent de nombreuses informations utiles pour leur partitionnement ultérieur. Nous listons ici quelques champs remarquables transmis par les RS.

3.1.1 Adresse MAC

Afin que les communications WiFi entre les appareils et les fournisseurs d'accès n'interfèrent pas entre elles, il est nécessaire d'utiliser un identifiant unique dans tous les paquets de données échangés. Cet identifiant est appelé adresse de "Media Access Control" (MAC).

L'adresse MAC est supposée être une adresse unique pour chaque périphérique dans le monde. Dans le protocole IEEE 802, elle est composée de 6 octets et généralement représentée par douze chiffres hexadécimaux. Les trois premiers octets sont appelés "Organizationally Unique

Identifier" (OUI) et sont vendus par l'IEEE aux fabricants d'appareils électroniques. Chaque fabricant dispose d'une plage d'OUI qui lui est dédiée. Chaque RS contient toujours deux champs qui sont des adresses MAC :

— L'adresse de l'émetteur : supposée être l'adresse MAC unique de l'appareil émettant la RS.
— L'adresse de destination : l'adresse du fournisseur d'accès visé par la RS s'il y en a un, ou une adresse par défaut si la RS n'est pas dirigée.

Antérieurement à 2013 les Requêtes de Sonde contenaient systématiquement l'adresse MAC unique de l'émetteur ce qui rendait leur suivi aisé. En effet, compter les appareils présents aux alentours dans un rayon d'environ 100m revient alors simplement à compter les différentes adresses MAC reçues sur une période donnée.

Un mécanisme de randomisation de l'adresse MAC de l'émetteur a depuis été implémenté par la plupart des fabricants de téléphones et développeurs de systèmes d'exploitation. Aujourd'hui, ce mécanisme est exploité par 50 à 80% des appareils utilisant le WiFi (Fenske et al. (2021)). Si l'on ne considère que les téléphones portables, la totalité de ceux que nous avons testés employaient la randomisation de l'adresse MAC, et d'autres modèles récents testés par Fenske et al. (2021) montraient le même comportement.

Cependant, afin de ne pas interférer avec les autres appareils, les adresses générées ne sont pas totalement aléatoires. En effet, un bit de l'adresse appelé "Universal/local bit" (U/L bit) permet de signaler que l'appareil émettant n'a pas utilisé son adresse MAC unique.

La Figure 3.1 montre un exemple d'adresse MAC et sa structure. Dans cet exemple, le bit U/L de l'adresse prend la valeur 1, ce qui signifie que c'est une adresse locale et non l'adresse unique de l'émetteur. Notons que pour une adresse randomisée, l'OUI utilisé est lui aussi généralement rendu aléatoire à l'exception du bit U/L.

L'existence du bit U/L nous permet, dans toute la suite, de distinguer les Requêtes de Sonde ayant une adresse MAC aléatoire de celles identifiées de manière unique, ces dernières ne nécessitant pas plus de travail pour être regroupées par appareil émettant. Cette différence d'approche selon le type d'adresse MAC est illustrée dans la figure 3.2.

3.1.2 SSID

Le "Service Set Identifier" (SSID) est un champ qui contient le nom du réseau que l'appareil essaie de découvrir s'il est configuré pour s'y connecter automatiquement. Souvent, il s'agit du nom du réseau domestique du propriétaire de l'appareil, ce qui peut dans certain cas rendre l'utilisateur identifiable et porter atteinte à sa vie privée.

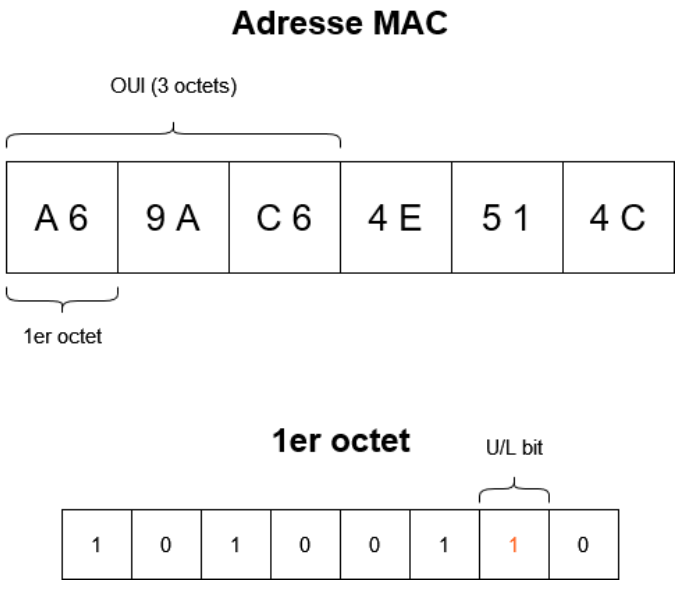

Figure 3.1 Exemple d'adresse MAC locale.

Cependant, les fabricants ont conscience de ce risque d'atteinte à la vie privée et ce champ est en conséquence rarement non vide (5% des Requêtes de Sonde seulement dans nos données collectées en 2021).

Le SSID est parfois utilisé car il reste le seul moyen de découvrir les réseaux cachés, c'est à dire les réseaux qui ignorent les Requêtes de Sonde si elles ne les ciblent pas explicitement.

Puisqu'il contient potentiellement un nom de réseau domestique unique, il peut s'avérer très précieux pour l'évaluation de la similitude entre deux RS.

3.1.3 RSSI

Le "Received Signal Strength Indicator" (RSSI) est un autre champ calculé au moment de la collecte des Requêtes de Sonde. Ce champ, en dB, donne une indication de la puissance du signal reçu. Il est cependant bien connu qu'il est peu corrélé avec la distance réelle entre l'émetteur et le récepteur (Longo et al. (2018)) en raison des nombreuses réflexions, diffractions et interférences qui peuvent se produire avec les ondes. De plus tous les appareils n'émettent pas à la même puissance.

Cependant, un appareil bougeant peu aura tendance à avoir un RSSI relativement stable, le rendant tout de même utile pour l'évaluation de la similarité entre deux RS.

3.1.4 Numéro de Séquence

Le "Numéro de Séquence" est un nombre compris entre 0 et 4095, cyclique et habituellement strictement incrémental. Il permet de vérifier que les paquets de données ont été reçus dans le bon ordre et qu'aucun n'est manquant.

Cependant, ce nombre a peu d'intérêt dans le contexte des Requêtes de Sonde puisque ces paquets ne servent pas à échanger des données. Une récente tendance des fabricants est de rendre ce nombre aléatoire lors de la construction des Requêtes de Sonde. La plupart des téléphones fabriqués après 2018 utilisent ce mécanisme (Fenske et al. (2021)).

Le numéro de Séquence était notamment utilisé par Vanhoef et al. (2016) au sein de leur méthode de regroupement des RS par téléphone d'origine. Le contexte actuel de randomisation de ce nombre rend leur approche obsolète. C'est une des raisons pour laquelle nous proposons une modification de leur algorithme au Chapitre 5 qui traite des algorithmes de partitionnement.

3.1.5 Éléments d'information

On appelle "Éléments d'information" (EI) l'ensemble des champs catégoriques. Il s'agit d'informations diverses utiles au protocole 802.11. Un exemple de ces éléments d'information est la liste des fréquences de transmissions supportées par l'appareil. Quelques EI sont donnés en exemple dans le Tableau 3.1. Une liste plus exhaustive et des explications plus détaillées peuvent être trouvées dans le livre de David A. Westcott (2011).

Dans les travaux de Vanhoef et al. (2016), ces champs sont supposés immuables et utilisés pour calculer la "signature" des Requêtes de Sonde, information ensuite utilisée pour le regroupement des Requêtes de Sonde.

3.2 Collecte de Requêtes de Sonde et données existantes

Notre travail est avant tout une approche théorique de modélisation. La difficulté d'obtenir suffisamment de données récentes nous force à nous reposer majoritairement sur des données publiques. Cela signifie que la validation de ce travail avec des appareils récents et les protocoles de communication WiFi de 2022 est limitée.

En effet, pour élaborer notre approche, nous nous reposons en majeure partie sur les données publiques de la base CRAWDAD-sapienza (Barbera et al. (2013)). Ces Requêtes de Sonde sont collectées dans différents lieux en Italie et sont antérieures à 2013, ce qui a deux conséquences :

Tableau 3.1 Quelques-uns des principaux Éléments d'Information utilisés dans les Requêtes de Sonde

EI	Définition
rates	Fréquences de transfert supportées
vendorspecific_info	Informations spécifiques du vendeur
info	Ensemble de champs réservés pour de futurs changements du protocole 802.11 mais parfois tout de même transmis
A_MSDU	Un premier type d'agrégation de paquets
DSSS_CCK	Support du type de modulation
Short_GI_20Mhz	Contrôle de l'intervalle entre la transmission de bits afin de minimiser les pertes par auto-interférences
SM_Power_Save	Support de la réduction d'antennes actives pour économiser de l'énergie
A_MPDU	Un autre type d'agrégation de paquets
MCS	Paramètres de la modulation

— Les adresses MAC transmises dans ces Requêtes de Sonde (bien qu'anonymisées dans le jeu de données) sont des identifiants uniques de l'appareil émetteur. Ce jeu de données est donc annoté pour notre cas d'étude.

— Le protocole 802.11 a évolué depuis 2013, notamment avec l'introduction de la randomisation des adresses MAC. Il est probable que des méthodes uniquement fondées sur les données de CRAWDAD nécessitent des ajustements afin de toujours fonctionner aujourd'hui.

Cependant, nous avons aussi à notre disposition un faible nombre de téléphones dont la liste des modèles est donnée dans le tableau 3.2. Ces appareils ont été choisis afin de représenter les fabricants ayant la part de marché la plus importante au Canada, ainsi que les différents systèmes d'exploitation existants. Nous collectons des Requêtes de Sonde en provenance de ces 5 téléphones et utilisons ces données pour vérifier qu'elles n'infirment pas nos modèles. Cependant, ces téléphones sont en nombre insuffisant pour valider complètement notre approche.

La collecte des données de ces 5 téléphones est faite à partir d'une carte WiFi réglée en "mode surveillance" et pilotée par un automate de type Nvidia Jetson Nano. La carte permet de recevoir des RS jusqu'à une portée de 100m environ en extérieur (Zhou et al. (2020)). L'automate permet le traitement des données et l'implémentation de nos algorithmes.

Tableau 3.2 Liste des modèles de téléphones mis à notre disposition

Modèle	Année	Système d'exploitation
Huawei P3	2019	Android
IPhone SE	2016	iOS
Google Pixel XL	2016	Android
Samsung Galaxy A51	2019	Android
Samsung Galaxy S8 P3	2017	Android

3.3 Conséquences et approche générale

La structure des RS est particulièrement importante pour la réalisation de notre premier objectif : l'établissement d'une métrique donnant la similarité entre les RS. Nous remarquons que les informations transmises par les RS sont de différentes natures (catégoriques et numériques) et doivent être traitées en conséquence.

Par exemple, deux adresses MAC identiques indiquent généralement que les RS viennent du même émetteur, mais deux adresses MAC différentes ne permettent généralement pas d'affirmer que les RS proviennent de deux appareils différents. Certains EI au contraire prennent peu de valeurs différentes mais sont très stables, permettant de distinguer deux émetteurs en cas de différence du champ.

Cependant, lors de l'élaboration de notre approche, il est important de noter que le protocole 802.11 est en constante évolution. Par exemple, de nouvelles fréquences de transmission peuvent être introduites, d'autres abandonnées. Ainsi, il est nécessaire d'élaborer une méthode flexible qui ne sera pas perturbée par des valeurs n'existant pas dans les données à notre disposition. De même, tous les champs ne sont pas systématiquement présents dans les RS. Les champs absents ne doivent pas non plus perturber notre approche.

Étant données les propriétés du protocole 802.11 exposées dans ce début de chapitre, nous donnons en Figure 3.2 notre approche générale pour atteindre notre objectif. Cette approche traite séparément les RS avec une adresse MAC locale et celles avec une adresse MAC globale. Dans le cas des adresses MAC globales, le regroupement se fait directement en utilisant l'adresse comme identifiant unique de l'appareil émettant. Dans le cas des adresses locales, nous sélectionnons d'abord les champs donnant le plus d'information, avant d'élaborer une métrique prenant en compte ces champs pour calculer la distance deux à deux entre les RS. Enfin, cette distance sera utilisée pour le partitionnement.

La suite du chapitre contient une analyse des données des RS de CRAWDAD et propose des

critères permettant de sélectionner les champs à utiliser pour notre approche.

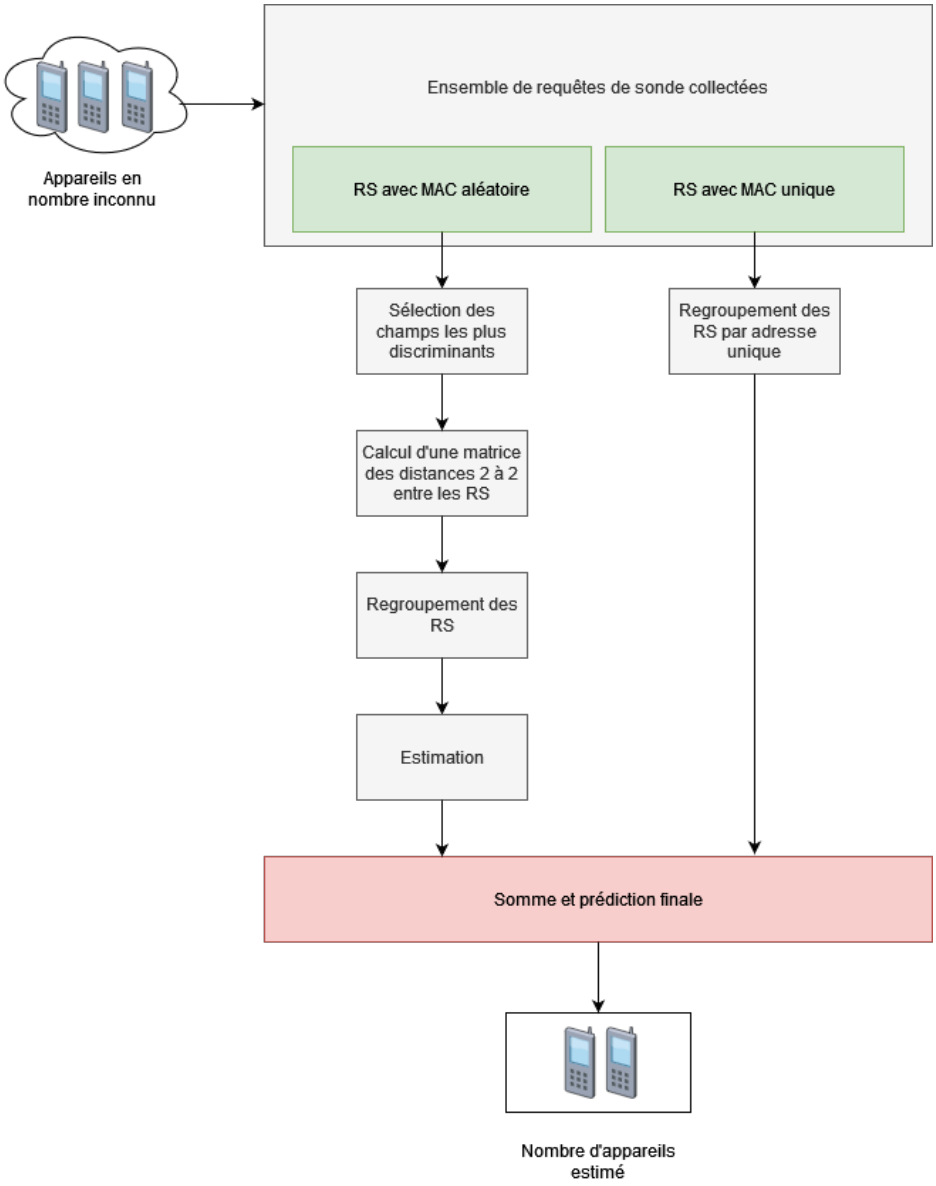

Figure 3.2 Récapitulatif de l'approche générale selon le type d'adresse MAC (globale ou locale)

3.4 Analyse des données

Les RS après décodage par le dispositif d'acquisition contiennent de très nombreux champs. Nous en avons identifié plus de 300 différents. Ces champs appartiennent à deux catégories :

— Les informations de capture, telles que l'heure d'arrivée de la Requête de Sonde, ou le RSSI. Ce sont des informations ajoutées par le dispositif de capture.
— Les informations du protocole 802.11, telles que les adresses MAC, SSID ou encore les fréquences supportées.

Cependant, tous ces champs ne sont pas équivalents. Beaucoup d'entre eux sont en fait généralement absents des RS et certains prennent un nombre très restreint de valeurs. L'objectif de cette section est d'établir des critères permettant d'évaluer l'information donnée par chaque champ pour ensuite ne retenir que les champs les plus utiles à notre cas d'étude.

3.4.1 Notations

Nous introduisons les notations suivantes :

t	Un téléphone
p	Une RS
F	Un champ (représente un des éléments d'information des RS)
F_p	Le champ F de la RS p
Ω_F	L'ensemble des valeurs possibles pour le champ F
$H(F)$	L'entropie de Shannon associée au champ F
$S(F)$	La stabilité du champ F

3.4.2 Présence

On appelle présence du champ F la probabilité qu'une RS contienne un champ F non vide. C'est à dire :

$$\mathbb{P}[F \neq \emptyset].$$

En effet, la plupart des champs que nous avons identifiés ne sont pas transmis dans l'immense majorité des RS. Dans toute la suite, nous traitons $\emptyset$ comme une valeur possible à part entière du champ F.

3.4.3 Pouvoir de discrimination

On appelle pouvoir de discrimination du champ F la probabilité que deux RS p_1 et p_2 proviennent de téléphones différents ($t_1 \neq t_2$) sachant que leur champ F est différent :

$$D(F) = \mathbb{P}[t_1 \neq t_2 | F_{p_1} \neq F_{p_2}].$$

C'est idéalement cette quantité que l'on souhaiterait utiliser pour sélectionner les champs pour notre approche. Cependant, cette probabilité se trouve être très difficile à évaluer à partir de nos données. En effet, $D(F)$ dépend notamment du nombre de téléphones concurrents présents au moment de la capture.

Nous définissons donc deux autres grandeurs, indépendantes du nombre de téléphones présents au moment de la capture, pour sélectionner les champs les plus discriminants et nous affranchir de cette difficulté. Il s'agit de la stabilité et de l'entropie de Shannon.

3.4.4 Stabilité

La stabilité correspond à la tendance qu'a un champ à prendre la même valeur entre deux RS émises par un même téléphone. C'est à dire pour un champ F et deux RS p_1 et p_2 provenant du même téléphone ($t_1 = t_2$) :

$$S(F) = \mathbb{P}[F_{p_1} = F_{p_2} | t_1 = t_2].$$

Pour être mesurée, la stabilité nécessite de savoir de quel téléphone proviennent les RS. On doit donc faire cette mesure soit sur les données annotées (mais périmées) de CRAWDAD, soit sur les données issues des captures individuelles des téléphones à notre disposition (en nombre très limité).

Ainsi, un champ avec une grande stabilité, sous réserve qu'il prend également une grande diversité de valeurs, pourra s'avérer crucial pour notre étude. La Figure 3.3 donne quelques-unes de ces stabilités en abscisse.

3.4.5 Entropie de Shannon

Pour chaque champ F, on peut définir l'entropie de Shannon :

$$H(F) = -\sum_{x \in \Omega_F} \mathbb{P}[F = x] \ln(\mathbb{P}[F = x]).$$

Cette grandeur est une mesure de la quantité d'information contenue dans un champ. Si l'on remplace le logarithme naturel par un logarithme en base 2, elle correspond à une borne inférieure du nombre de bits nécessaires en moyenne pour coder le champ en utilisant un schéma de codage optimal.

Par exemple, un champ ne prenant qu'une valeur avec probabilité 1 a une entropie de $H = 0$, ce qui signifie qu'il n'apporte aucune information pour distinguer les RS. Si deux valeurs peuvent être prises avec probabilité $\frac{1}{2}$, l'entropie en base 2 est $H = -2 \times \frac{1}{2} \log_2(\frac{1}{2}) = 1$ ce qui signifie qu'un unique bit est suffisant pour représenter l'information.

La Figure 3.3 donne quelques-unes de ces entropies en ordonnée pour certains champs.

3.5 Méthode de sélection des champs

L'entropie de Shannon permet d'évaluer la quantité d'informations qu'un champ donne pour discriminer deux RS entre elles. Cependant, si elle est associée à une trop faible stabilité, cela diminue son utilité pour évaluer si les RS comparées proviennent du même téléphone. C'est pourquoi il est nécessaire d'utiliser une combinaison de ces deux métriques pour décider quels sont les champs à retenir.

Un compromis possible est de trier les champs par $S(F) \times H(F)$ décroissant, puis de retenir les n meilleurs champs selon ce critère. On choisit dans ce critère de donner le même poids à l'entropie et à la stabilité, car des travaux précédents comme ceux de Vanhoef et al. (2016) accordent la même importance à ces deux grandeurs. Le plus important est, qu'à entropie égale, les champs de plus grande stabilité soient privilégiés. De même, à stabilité égale, les champs de plus grande entropie sont privilégiés. Cependant, il est important de noter que le choix de cette combinaisons de critères reste largement arbitraire.

La Figure 3.3 donne la répartition des champs selon le critère $S(F) \times H(F)$. Chaque point représente un champ. Le gradient de couleur donne la valeur de $S(F) \times H(F)$. Le nom des champs est affiché lorsque la lisibilité le permet.

Notons que les valeurs d'entropies et de stabilité ont été calculées avec les données de CRAWDAD. Cela explique par exemple que le champ "sender_addr" (qui est l'adresse MAC de l'émetteur) est le meilleur champ. En effet, il s'agit d'un identifiant unique des téléphones dans cette base de données. La sélection des champs sur CRAWDAD permet également de s'assurer que le critère $S(F) \times H(F)$ n'est pas influencé par les comportements de randomisation, qui sont postérieurs à 2013.

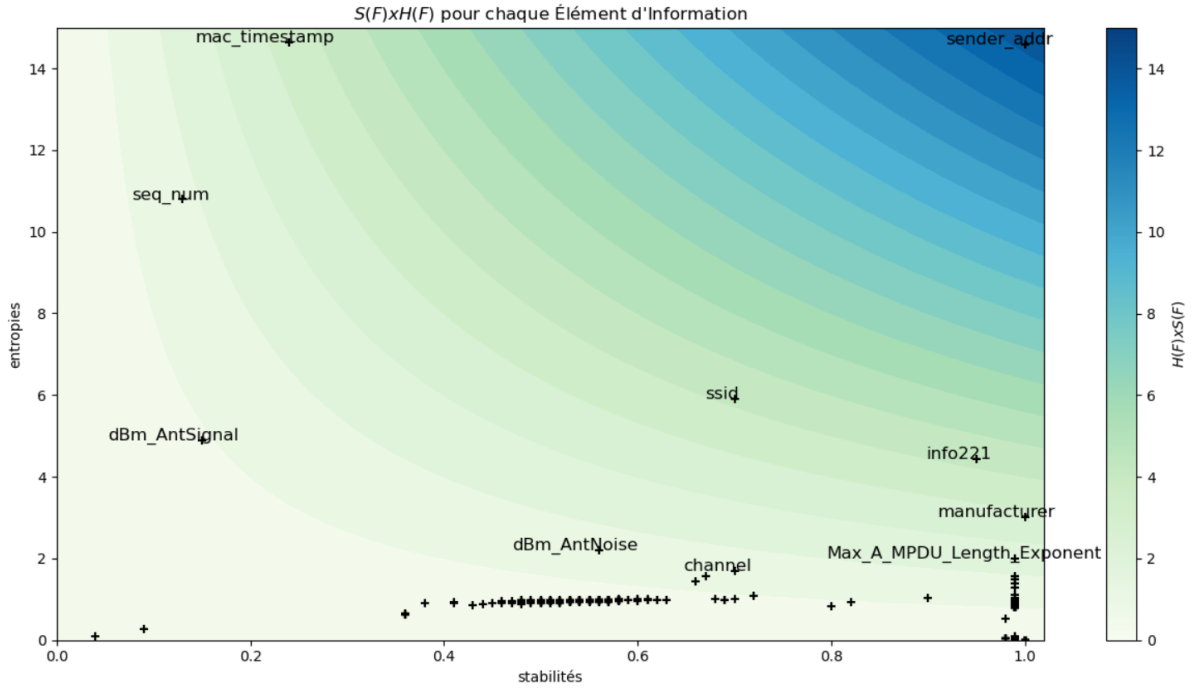

Figure 3.3 Champs selon le critère $S(F) \times H(F)$

3.6 Conclusion et critère de sélection retenu

Dans ce chapitre nous avons défini une méthode de sélection des champs des RS donnant potentiellement le plus d'information pour les discriminer. Notre analyse nous suggère de retenir les champs ayant à la fois une haute entropie et une haute stabilité. Nous avons retenu le critère $H(F) \times S(F)$ décroissant.

Dans la suite, les champs sélectionnés par ce critère seront utilisés afin de définir une métrique sur l'espace des Requêtes de Sonde et permettre leur regroupement.

CHAPITRE 4 MÉTRIQUE SUR L'ESPACE DES REQUÊTES DE SONDE

Les algorithmes de partitionnement sont des algorithmes permettant de décomposer un ensemble en un certain nombre de sous-ensembles. Pour fonctionner, ces algorithmes nécessitent une métrique pour comparer deux à deux les éléments que l'on souhaite regrouper. Lorsque les données sont des nombres réels (ou des vecteurs de nombre réels), une approche simple est la suivante :

— Analyse en composantes principales et changement de base associé pour réduire la dimensionnalité des données et les normaliser.

— Utilisation de la distance euclidienne pour évaluer la similarité entre deux points de données.

Cependant, dans notre cas, les données ne sont pas des vecteurs de nombres réels, mais un mélange entre une majorité de données catégoriques (les Éléments d'Information) et quelques données numériques (le temps d'arrivée, le RSSI).

Dans ce chapitre, nous développons une métrique hybride permettant de prendre en compte à la fois les données catégoriques et les données scalaires dans l'évaluation de la similitude entre deux RS.

4.1 Métriques existantes pour les données catégoriques

Pour les données catégoriques, la distance euclidienne ne peut pas être directement utilisée. Il existe cependant d'autres distances exploitables. Nous en présentons deux exemples : la distance de Hamming et l'indice de Jaccard.

4.1.1 Distance de Hamming

Cette distance consiste simplement à compter le nombre de champs qui ne sont pas identiques entre deux RS. Avec nos notations, si p_1 et p_2 sont deux RS et $(F^i)_{0\leq i<n}$ sont n champs catégoriques, la distance de Hamming s'exprime de la façon suivante :

$$d_{Hamming}(p_1, p_2) = \sum_{i=0}^{n-1} 1_{F^i_{p_1} \neq F^i_{p_2}}.$$

Avec la notation suivante :

$$1_{F^i_{p_1} \neq F^i_{p_2}} = \begin{cases} 0 & \text{si } F^i_{p_1} = F^i_{p_2} \\ 1 & \text{sinon.} \end{cases}$$

Cette distance est simple et rapide à calculer, mais elle n'introduit aucune hiérarchie entre les champs. Or, l'analyse de données menée dans le chapitre précédent nous informe que certains champs apportent beaucoup plus d'informations que d'autres, et devraient donc être pris en compte différemment.

4.1.2 Indice de Jaccard

Cet indice est très similaire à la distance de Hamming. Cependant, en plus de tenir compte du nombre de champs différents, il tient également compte du nombre total de champs. Il est ainsi adapté à des entrées de dimensions différentes, et peut être défini de façon à ne considérer que les champs non vides.

Cet indice est normalement défini de façon à ce que sa valeur soit proche de 1 si les RS sont similaires, et de 0 sinon. Nous proposons ici une définition différente que l'on appelle "distance de Jaccard" et qui tend au contraire à être proche de 0 pour des RS similaires et plus grand pour des RS différentes.

Ainsi, étant donné deux RS p_1 et p_2 et si $(F^i)_{0 \leq i < n^*}$ est l'ensemble des champs tels que $\forall i \in [\![0, n^*[\![$, $F^i_{p_1}$ ou $F^i_{p_2}$ est non vide, alors nous définissons la distance de Jaccard de la façon suivante :

$$d_{Jaccard}(p_1, p_2) = \frac{1}{n^*} \sum_{i=0}^{n^*-1} 1_{F^i_{p_1} \neq F^i_{p_2}}.$$

Cette distance est également simple et rapide à calculer. Néanmoins, elle n'introduit pas non plus de hiérarchie entre les différents champs. Avec cette définition, la distance de Jaccard revient simplement à normaliser la distance de Hamming en la divisant par le nombre de champs non vides. Intuitivement, cela revient à remplacer le compte direct du nombre de champs différents par le taux de champs différents.

Dans la partie suivante, nous allons nous inspirer de ces métriques pour mettre au point une fonction de distance plus adaptée à ce que l'on sait des RS.

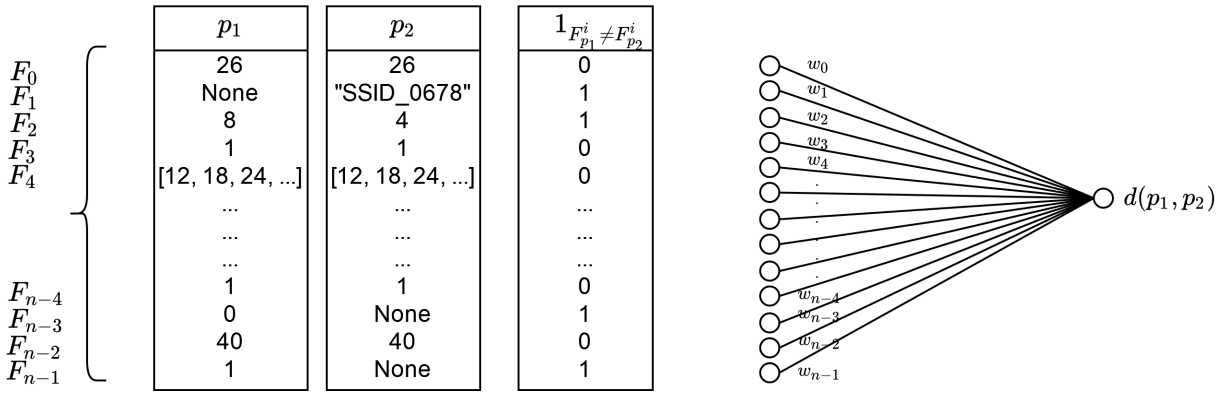

Figure 4.1 Schématisation de la fonction de distance

4.2 Proposition d'une nouvelle métrique pour les données catégoriques de l'espace des RS

Tel qu'expliqué précédemment, les métriques de Hamming et de Jaccard n'introduisent aucune hiérarchie entre les champs. Or, certains champs donnent des informations bien plus significatives que d'autres.

Par exemple, l'Élément d'Information appelé "manufacturer" a été mesuré avec une stabilité de 100% sur les données de CRAWDAD. Ainsi, si deux RS ont un champ "manufacturer" différent, on peut affirmer avec certitude qu'elles proviennent de téléphones différents.

En revanche, le champ appelé "SSID" n'a qu'une stabilité de 70%. Ainsi deux RS ayant un champ "SSID" différent ne proviennent pas forcément de deux appareils différents.

Nous proposons donc de pondérer la distance de Hamming pour prendre en compte cette différence entre les champs.

Nous définissons donc la distance suivante que l'on nomme "distance de Hamming pondérée" :

$$d(p_1, p_2) = \sum_{i=1}^{n} w_i 1_{F^i_{p_1} \neq F^i_{p_2}}.$$

La figure 4.1 représente schématiquement cette fonction de distance.

Nous allons dans un premier temps discuter du choix des w_i, avant de démontrer quelques propriétés de cette nouvelle métrique.

4.2.1 Critère de choix des w_i

Soient deux RS p_1 et p_2 provenant respectivement des téléphones t_1 et t_2 (avec éventuellement $t_1 = t_2$ si les RS viennent du même téléphone).

Les w_i doivent être choisis selon le critère suivant :

- Si $t_1 = t_2$, $d(p_1, p_2)$ doit être proche de 0.
- Si $t_1 \neq t_2$, $d(p_1, p_2)$ doit être grand, par exemple supérieur à 1. Nous choisissons 1 arbitrairement, uniquement pour introduire une séparation entre les exemples pour lesquels $t_1 = t_2$ et ceux pour lesquels $t_1 \neq t_2$.

On définit ainsi une fonction de perte, qui combine ces deux critères :

$$L = \begin{cases} d(p_1, p_2)^2 & \text{si } t_1 = t_2 \\ \max\{0, 1 - d(p_1, p_2)\}^2 & \text{sinon.} \end{cases}$$

On peut alors réécrire le problème du choix des w_i comme un problème d'optimisation. On cherche les poids w_i qui minimisent l'espérance de la fonction de perte $\mathbb{E}[L]$.

Ce problème d'optimisation est difficile pour deux raisons :

- La forme inhabituelle de la fonction de perte choisie. En effet, cette fonction fait intervenir un maximum afin de ne pas pénaliser les distances supérieures à 1 dans le cas $t_1 \neq t_2$. Une fonction de perte plus classique comme l'erreur quadratique moyenne aurait permis de trouver ces coefficients par une simple régression linéaire.
- La contrainte $\forall i, w_i > 0$ que l'on montrera nécessaire au respect de l'inégalité triangulaire dans la section suivante.

Nous proposons donc d'employer des méthodes d'apprentissage machine pour estimer la solution de ce problème d'optimisation.

4.2.2 Apprentissage machine

Nous allons donc recourir à l'apprentissage machine pour déterminer des coefficients w_i adaptés. Nous utilisons le langage Python et la librairie Tensorflow pour appliquer la descente de gradient stochastique. Il s'agit d'un algorithme qui permet d'ajuster les paramètres d'un modèle en calculant les dérivées partielles de la fonction de perte par rapport aux paramètres (ici les poids w_i).

Tout d'abord, nous utilisons la base de données annotée CRAWDAD afin de générer deux type d'exemples :

- Des exemples positifs, pour lesquels nous souhaitons une distance proche de 0. On

pose $d_{vraie} = 0$ pour ces exemples.
— Des exemples négatifs, pour lesquels nous souhaitons une distance supérieure à 1. On pose $d_{vraie} = 1$ pour ces exemples.

Nous réécrivons également la fonction de perte de façon matricielle afin de vectoriser (et donc d'accélérer) les calculs :

$$L = [d_{vraie} \times \max\{d_{vraie} - d(p_1, p_2), 0\} + (1 - d_{vraie}) \times (d_{vraie} - d(p_1, p_2))]^2.$$

Nous ajoutons ensuite la contrainte $\forall i, w_i > 0$, et nous appliquons la descente de gradient stochastique jusqu'à convergence.

Ainsi, nous obtenons des coefficients w_i minimisant au moins localement la moyenne arithmétique de la fonction de perte $\bar{L}$ (qui approche l'espérance $\mathbb{E}[L]$). Notons également que cette approche est généralisable et qu'elle permet d'élaborer des fonctions de distance plus complexes utilisant par exemple plusieurs couches cachées. Cependant, choisir une fonction de cette forme permet d'assurer quelques propriétés que nous détaillons dans la section suivante.

4.3 Propriétés

Dans cette section, nous démontrons que la fonction $d(p_1, p_2) = \sum_{i=1}^{n} w_i 1_{F^i_{p_1} \neq F^i_{p_2}}$ a bien les propriétés d'une distance, à condition que $\forall i, w_i > 0$. Nous montrons également que la métrique ainsi définie est plus performante que la distance de Hamming au sens de la moyenne arithmétique de la fonction de perte $\bar{L}$. Enfin, nous montrons l'existence d'une distance critique permettant d'affirmer que deux RS proviennent de téléphones différents.

4.3.1 Propriétés des distances

LEMME 1

La fonction définie par $d(p_1, p_2) = \sum_{i=1}^{n} w_i 1_{F^i_{p_1} \neq F^i_{p_2}}$ est une distance sur l'espace des RS à condition que $\forall i, w_i > 0$. C'est à dire qu'elle est :
— Symétrique, $d(p_1, p_2) = d(p_2, p_1.)$
— Séparable, $d(p_1, p_2) = 0 \iff p_1 = p_2$.
— Elle respecte l'inégalité triangulaire, $\forall p_1, p_2, p_3, d(p_1, p_3) \leq d(p_1, p_2) + d(p_2, p_3)$.

Démonstration :

— On a $\forall i, w_i 1_{F^i_{p_1} \neq F^i_{p_2}} = w_i 1_{F^i_{p_2} \neq F^i_{p_1}}$. Donc $d(p_1, p_2) = d(p_2, p_1)$. Cela démontre la symétrie.

— On a $d(p_1, p_2) = 0$ si et seulement si $\forall i, F^i_{p_1} = F^i_{p_2}$ par stricte positivité des w_i. C'est à dire si et seulement si $p_1 = p_2$. Cela démontre la séparabilité.

— On remarque que $\forall i, 1_{F^i_{p_1} \neq F^i_{p_3}} \leq 1_{F^i_{p_1} \neq F^i_{p_2}} + 1_{F^i_{p_2} \neq F^i_{p_3}}$. L'inégalité triangulaire en découle par multiplication par $w_i > 0$ et par sommation.

■

4.3.2 Supériorité à la distance de Hamming sous réserve de convergence

La propriété suivante est vraie sous réserve de convergence de la dexcente de gradient stochastique. On considère qu'il y a convergence lorsque la fonction de distance atteint un minimum local de la moyenne arithmétique de la fonction de perte. Une condition nécessaire pour cela est $\forall i, \dfrac{\partial \bar{L}}{\partial w_i} = 0$. En pratique on considère qu'il y a convergence lorsque $\bar{L}$ ne diminue plus lors de la descente de gradient stochastique.

PROPRIÉTÉ 1

La distance de Hamming pondérée a une fonction de perte plus basse en moyenne que la distance de Hamming.

Démonstration :

Nous remarquons que la distance de Hamming est un cas particulier de la distance pondérée où $\forall i, w_i = 1$. On en déduit qu'en initialisant les poids w_i à 1, le modèle initial est la distance de Hamming. On en déduit que sous réserve de convergence, $\bar{L}_{Hamming} \geq \bar{L}_{coeffs}$.

■

Remarque :

Il n'est pas nécessaire d'avoir convergence vers un minimum global pour obtenir ce résultat en raison de l'initialisation particulière des w_i. Cependant, en pratique, on obtient de meilleurs résultats avec une initialisation aléatoire des poids.

4.3.3 Existence d'une distance critique pour la provenance du même téléphone

LEMME 2

Il existe une distance critique, notée ϵ^*, telle que $t_1 = t_2 \implies d(p_1, p_2) \leq \epsilon^*$.

Démonstration :

Il suffit de remarquer que $\forall p_1, p_2, d(p_1, p_2) \leq \sum_{i=1}^{n} w_i$. On peut donc choisir $\epsilon^* = \sum_{i=1}^{n} w_i$.

■

Remarque :

En pratique on peut trouver ϵ^* relativement petit. Cela signifie que dans certains cas il est possible d'affirmer avec certitude (par contraposition) que $t_1 \neq t_2$ simplement en vérifiant que $d(p_1, p_2) > \epsilon^*$.

Plus tard, cette propriété sera utilisée pour construire un estimateur du nombre d'appareils détectés.

Remarquons également que l'inverse n'est pas vrai, il n'existe pas $\epsilon > 0$ tel que $t_1 \neq t_2 \implies d(p_1, p_2) > \epsilon$ car certaines RS peuvent être identiques sans provenir du même téléphone.

4.4 Prise en compte des champs numériques

Dans la partie précédente, nous avons mis au point une métrique prenant en compte les champs catégoriques des RS. Le but de cette section est de montrer comment généraliser cette métrique pour également prendre en compte les champs numériques.

4.4.1 Approche

De la même façon que nous avons fait intervenir la fonction indicatrice $1_{F^i_{p_1} \neq F^i_{p_2}}$ pour comparer les champs catégoriques des Requêtes de Sonde, nous proposons d'également utiliser la différence absolue entre la valeur de certains champs numériques comme entrée pour notre métrique.

Deux champs numériques semblent intéressants à prendre en compte :

— Le temps d'arrivée des RS. On note $|T_{p_2} - T_{p_1}|$ l'intervalle entre les RS.

— La variation du RSSI dans le temps. On définit cette quantité comme $|\frac{RSSI_{p_2} - RSSI_{p_1}}{T_{p_2} - T_{p_1}}|$.

En ce qui concerne le RSSI, une attention particulière est à porter dans le cas où T_{p_1} et T_{p_2} sont très proches. Ce cas est problématique car la valeur de $|\frac{RSSI_{p_2} - RSSI_{p_1}}{T_{p_2} - T_{p_1}}|$ peut exploser pour des RS émises par le même téléphone à des intervalles très courts. On propose d'utiliser la différence de RSSI directement pour éviter ce problème.

La métrique coefficientée s'écrirait alors :

$$d(p_1, p_2) = \sum_{i=1}^{n} w_i 1_{F^i_{p_1} \neq F^i_{p_2}} + w_{n+1}|T_{p_2} - T_{p_1}| + w_{n+2}|RSSI_{p_2} - RSSI_{p_1}|.$$

On peut calculer les coefficients (w_i) de la même façon que précédemment, en utilisant la descente de gradient stochastique.

4.4.2 Conservation des propriétés de la métrique

LEMME 3

La métrique hybride

$$d(p_1, p_2) = \sum_{i=1}^{n} w_i 1_{F^i_{p_1} \neq F^i_{p_2}} + w_{n+1}|T_{p_2} - T_{p_1}| + w_{n+2}|RSSI_{p_2} - RSSI_{p_1}|.$$

est une distance sur l'espace des RS.

Démonstration :

Pour montrer que l'on a toujours bien une distance sur l'espace des RS, il suffit de remarquer que :

- $\sum_{i=1}^{n} w_i 1_{F^i_{p_1} \neq F^i_{p_2}}$ est une distance sur $\prod_{i=1}^{n} F^i$ où $\prod$ est le produit cartésien.
- $|T_{p_2} - T_{p_1}|$ et $|RSSI_{p_2} - RSSI_{p_1}|$ sont des distances sur T et $RSSI$ respectivement en tant que norme 1 (où la norme 1 est définie comme la somme des modules des coefficients).

$d(p_1, p_2) = \sum_{i=1}^{n} w_i 1_{F^i_{p_1} \neq F^i_{p_2}} + w_{n+1}|T_{p_2} - T_{p_1}| + w_{n+2}|RSSI_{p_2} - RSSI_{p_1}|$ est donc bien une distance sur $RSSI \times T \times \prod_{i=1}^{n} F^i$ en tant que combinaison linéaire de distances (sous réserve que $\forall i, w_i > 0$).

■

4.5 Conclusion sur les métriques

Dans ce chapitre nous avons mis au point une nouvelle métrique permettant de comparer les RS deux à deux. Cette métrique a été optimisée afin de refléter l'appartenance des RS à un même appareil. Dans le chapitre suivant, nous allons exploiter cette métrique et ses propriétés pour tenter de regrouper les RS par appareil d'origine.

CHAPITRE 5 ALGORITHMES DE PARTITIONNEMENT

Maintenant que nous avons à notre disposition une métrique traduisant la similarité entre les Requêtes de Sonde, nous cherchons à les regrouper par appareil d'origine. Il existe de nombreuses approches pour effectuer ce regroupement. Nous en présentons deux et en proposons des modifications adaptées au contexte des RS.

5.1 Regroupement au fur et à mesure, approche de Vanhoef et al. renouvelée

La première approche que nous étudions est proposée par Vanhoef et al. (2016). Cette approche date de 2016 et est une des premières alternatives proposées à l'utilisation de l'adresse MAC pour le tracking des appareils connectés. Les auteurs démontrent dans cet article que dans la plupart des cas, les renseignements anonymes contenus par les Requêtes de Sonde sont suffisants pour suivre localement un téléphone.

L'algorithme original utilise des propriétés périmées des RS pour leur regroupement, nous en proposons donc une version renouvelée utilisant notre métrique nouvellement définie.

5.1.1 Algorithme original

Dans l'article de Vanhoef et al. (2016), l'algorithme 1 est proposé pour le regroupement des Requêtes de Sonde. Celui-ci traite les Requêtes de Sonde au fur et à mesure de leur arrivée pour les répartir dans des groupes immuables. Ce regroupement est fait sur la base de trois facteurs :

- L'égalité stricte des "signatures" ("fingerprints" en anglais) des Requêtes de Sonde. Il s'agit simplement de l'ensemble des valeurs des Éléments d'Information retenus.
- Une différence raisonnable entre les Numéros de Séquence (SN) de deux Requêtes de Sonde consécutives ($lastSN_{groupe} - SN_{p_n} < \Delta SN$).
- Une différence de temps d'arrivée raisonnable entre deux Requêtes de Sonde consécutives ($T_{p_n} - T_{p_{groupe}} < \Delta T$).

5.1.2 Problèmes avec l'algorithme original

Cet approche qui date de 2016 est très perturbée par les changements du protocole 802.11 depuis 2016.

Voici notamment quelques faiblesses de cet algorithme :

Algorithme 1 : Algorithme de regroupement de Vanhoef et al.

Entrées : P : Liste de Requêtes de Sonde capturées
ΔT : Temps maximum entre deux requêtes de Sonde
ΔSN : Différence maximum de numéro de séquence
Sortie : D : l'ensemble des groupes de Requêtes de Sonde

$M \leftarrow \emptyset$ // Regroupe les RS par signature
pour $p \in P$ **faire**
  $f \leftarrow$ signature(p)
  $M[f]$.ajouter(p)
fin

$D \leftarrow []$
pour $C \in M$ **faire**
  $S \leftarrow []$ // Contient les sous-groupes de M
  $m \leftarrow \max(p.SN$ pour p dans C)
  pour $p \in C$ **faire**
    trouver i tel que :
      d($p.SN$, $C.dernier\_SN$, m) $< \Delta SN$
      et $p.temps - C.dernier\_temps < \Delta T$
    si *i non trouvé* **alors**
      $i \leftarrow |S|$ // Création d'un nouveau groupe
    fin
    $S[i].ajouter(p)$
  fin
  $D.étendre(S)$ // On ajoute les sous-groupes à la liste de groupes
fin
retourner D

— Il est très vulnérable aux comportements de randomisation. En effet, l'algorithme requiert une égalité stricte entre les éléments d'information de deux RS pour pouvoir les considérer comme provenant du même téléphone. Or un téléphone rendant certains de ces champs aléatoires peut générer des RS avec des signatures très différentes, au point de s'étendre sur plusieurs dizaines voire centaines de groupes. Ce problème est déjà soulevé dans l'article de Vanhoef et al., mais aucune solution n'est proposée. L'algorithme a pour but original de traquer parfaitement les téléphones avec haute probabilité, ce qui diffère légèrement de notre objectif de comptage. Il est donc important pour nous de remédier à cette faiblesse.

— Une gestion périmée du Numéro de Séquence. Comme précisé dans le chapitre 3, le Numéro de Séquence est également rendu aléatoire pour la quasi totalité des téléphones produits depuis 2018. Cela rend obsolète toute la partie de l'algorithme qui repose sur cet attribut.

— Certains champs, comme le RSSI, ne sont pas exploités.

Nous proposons donc une version modifiée de l'algorithme pour tenter de remédier à ces faiblesses.

5.1.3 Algorithme modifié

Nous proposons de réadapter cet algorithme au contexte actuel. L'algorithme 2 est une généralisation de l'algorithme 1. Il introduit une fonction de distance utilisée pour comparer la similarité de deux Requêtes de Sonde. On peut montrer aisément que la fonction de distance donnée par l'algorithme 3 permet de retrouver l'algorithme initial.

Algorithme 2 : Algorithme de regroupement de Vanhoef et al. renouvelé

Entrées : P : Liste de Requêtes de Sonde capturées
ϵ : Une distance maximale pour le regroupement
Sortie : D : l'ensemble des groupes de Requêtes de Sonde

$D \leftarrow []$
pour $p \in P$ **faire**
  $d_{min} \leftarrow \min_{C \in D}\{distance(p, \text{dernière RS de C})\}$
  $i \leftarrow \underset{C \in D}{argmin}\{distance(p, \text{dernière RS de C})\}$
  si $d_{min} > \epsilon$ **alors**
    $i \leftarrow |D|$ // Création d'un nouveau groupe
  fin
  $D[i].ajouter(p)$ // Ajout au groupe i, ou création d'un nouveau groupe
fin
retourner D

L'introduction d'une fonction de distance permet l'utilisation de métriques plus complexes sur l'espace des Requêtes de Sonde. Notamment, celle que nous avons définie au chapitre précédent.

5.2 Regroupement direct, DBSCAN

L'algorithme DBSCAN ("Density-Based Spatial Clustering of Applications with Noise"), proposé par Ester et al. (1996), est un algorithme toujours très utilisé de nos jours pour le regroupement de données (Schubert et al. (2017)). Nous allons d'abord rappeler son fonctionnement, puis nous détaillerons le choix des paramètres.

Algorithme 3 : Fonction de distance de l'algorithme de Vanhoef et. al original

Entrées : p_2 une Requête de Sonde et C un groupe de Requêtes de Sonde
ΔT : Temps maximum entre deux Requêtes de Sonde
ΔSN : Différence maximum de numéro de Séquence
ϵ : La distance maximum de regroupement utilisée dans l'algorithme 2
Sortie : d : une distance

$p_1 \leftarrow C.dernière\_RS$
$m \leftarrow \underset{p \in C}{max}\{p.SN\}$
si $signature(p_1) \neq signature(p_2)$ **alors**
  retourner $\epsilon + 1$
sinon
  si $d(p_2.SN, p_1.SN, m) > \Delta SN$ **alors**
    retourner $\epsilon + 1$
  fin
  si $p_2.temps - p_1.temps > \Delta T$ **alors**
    retourner $\epsilon + 1$
  fin
fin
retourner 0

5.2.1 Fonctionnement

Cet algorithme utilise deux paramètres :

- ϵ, un rayon choisi pour représenter la distance maximale entre deux points de données appartenant au même groupe.
- *min_pts*, le nombre minimum de points nécessaires présents dans un rayon ϵ d'un point donné pour que celui-ci soit comme considéré suffisamment central pour fonder un groupe. Ces points centraux sont appelés "points coeurs".

L'algorithme calcule d'abord les points coeurs, puis évalue quels autres points sont atteignables à partir de ces points coeurs. Il les regroupe ensuite. Les points non atteignables sont traités comme des valeurs aberrantes.

Le fonctionnement complet de l'algorithme est détaillé par Ester et al. (1996). Dans la partie suivante, nous allons détailler le choix des paramètres les plus adaptés pour le cas des RS.

5.2.2 Choix des paramètres

Pour choisir *min_pts*, il est important de prendre en compte le nombre typique de RS envoyées par les téléphones dans le cas d'étude que l'on considère. L'intervalle de temps entre l'émission de deux Requêtes de Sonde par un même téléphone est aléatoire, et généralement

de l'ordre d'une à trois minutes. À chaque émission, nous avons observé que de 1 à 5 copies de la même RS sont enregistrées par notre dispositif de capture. Il serait donc envisageable d'utiliser $min\_pts = 6$ par exemple. Cela permettrait de potentiellement ignorer certains appareils dont le temps de présence est trop faible, et qui sont donc peu susceptibles d'être intéressants dans certains cas d'étude. L'algorithme DBSCAN considèrerait alors certains groupes de données comme étant des valeurs aberrantes.

Cependant, dans la suite du mémoire, nous nous concentrons uniquement sur le nombre de téléphones présents et ne souhaitons pas en exclure. Pour cette raison, le paramètre min_pts sera choisi égal à 1. Dans ce cas particulier, l'algorithme DBSCAN est équivalent à l'algorithme de partitionnement hiérarchique.

En ce qui concerne ϵ, nous choisissons une valeur légèrement supérieure à la distance critique ϵ^* pour la métrique définie dans le chapitre précédent. Cela permet d'obtenir la propriété que l'on nomme "Non duplication des groupes" et que l'on détaille dans la section suivante.

5.3 Propriétés des algorithmes de regroupement

Dans cette partie nous allons donner quelques propriétés des algorithmes de partitionnement. Nous allons montrer qu'il est possible de choisir leurs paramètres de façon à ce que le nombre de groupes créés lors du partitionnement soit une borne inférieure du compte réel.

Nous allons plus loin et proposons également une modélisation du nombre de groupes prédit en fonction d'une caractéristique de la métrique utilisée. Cette modélisation sera précieuse pour établir des intervalles de confiance sur la prédiction du nombre d'appareils connectés.

Enfin, nous allons comparer les complexités des deux algorithmes.

Dans toute la suite, le nombre de groupes formés lors du regroupement sera noté $\hat{N}$.

5.3.1 Non duplication des groupes

Une propriété préliminaire des partitionnements obtenus est celle de "Non duplication des groupes". C'est la tendance qu'ont les algorithmes à regrouper les RS d'un même téléphone dans le même groupe.

PROPRIÉTÉ 2

Il existe un paramètre ϵ tel que, quand utilisé dans l'algorithme DBSCAN ou l'algorithme de Vanhoef et. al renouvelé, le regroupement $(C_i)_{1\leq i\leq m}$ obtenu a la propriété suivante :

$$\forall (p_1, p_2) \text{ tels que } t_1 = t_2, p_1 \in C_k \implies p_2 \in C_k.$$

Autrement dit, les RS provenant du même téléphone sont regroupées dans le même groupe.

Démonstration :

On donne une démonstration très similaire à celle du lemme 2. Il suffit de choisir $\epsilon > \sum_{i=1}^{n} w_i$ qui est la distance maximale pour les métriques définies au chapitre précédent. Dans ce cas, qu'on utilise DBSCAN ou Vanhoef et al. renouvelé, toutes les RS d'un même téléphone sont regroupées dans un seul et unique groupe. Cela satisfait donc la propriété de non duplication des groupes.

■

Remarque : Il est souvent inutile de prendre ϵ très grand pour obtenir cette propriété. En fait, ϵ de l'ordre de ϵ^* (la distance critique pour la métrique utilisée) suffit souvent.

Cela se justifie par le fait que lors du partitionnement, deux erreurs peuvent se produire :

- Les RS de téléphones différents peuvent être regroupées par erreur. Cette erreur se produit lorsque des téléphones différents envoient des RS trop similaires. Cette erreur est inévitable avec nos métriques.
- Les RS d'un même téléphone peuvent être séparées par erreur. Même s'il est toujours possible de construire un contre-exemple, cette erreur a beaucoup moins de chance de se produire grâce à l'existence de la distance critique ϵ^*.

Ainsi si ϵ est légèrement supérieur à ϵ^*, les RS d'un même téléphone ont peu de chances d'être séparées lors du partitionnement.

On illustre ce phénomène dans la Figure 5.1. Sur cette figure, nous avons représenté trois RS en provenance de deux téléphones différents et leur partitionnement avec l'Algorithme 2. Les RS sont numérotées par ordre d'arrivée et p_1 et p_3 proviennent du même téléphone. Les RS sont représentées arbitrairement par des points sur un plan pour des raisons de schématisation. La distance euclidienne entre ces points représente la distance obtenue par une métrique quelconque. Dans les deux cas, une erreur est commise :

- Dans le premier cas, dit "typique", les RS sont si proches les une des autres qu'elles sont regroupées par erreur. Cependant, les RS du même téléphone sont groupées ensemble

et la propriété de non duplication des groupes est observée.

— Dans le second cas, p_2 est à la distance limite pour être regroupée avec p_1. Cela met exceptionnellement p_3 à une distance trop grande de p_2, la dernière RS du groupe, poussant l'algorithme à séparer p_1 et p_3. Ce cas est rendu rare en choisissant ϵ plus grand que la distance critique ϵ^*.

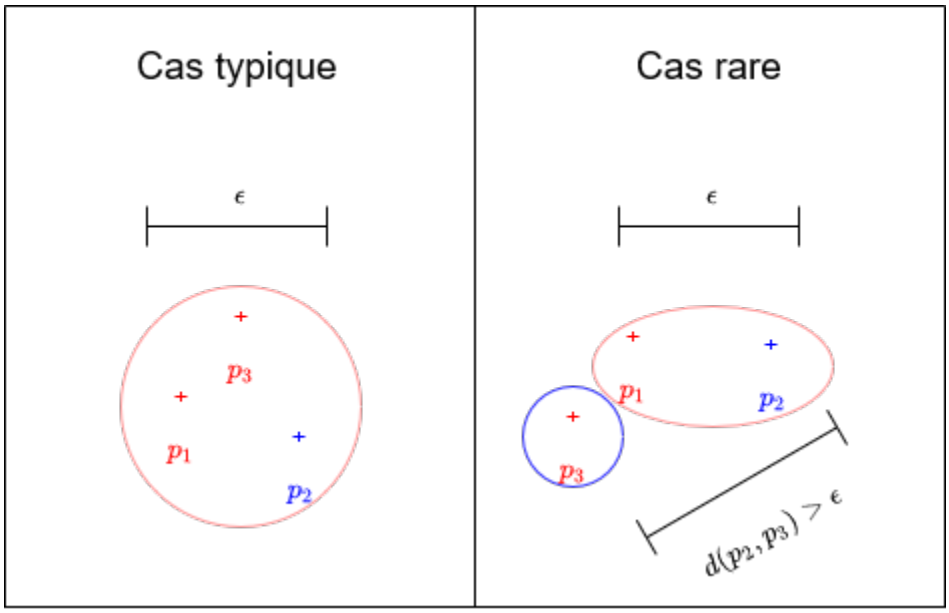

Figure 5.1 Illustration de la propriété de non duplication des groupes

Dans la partie suivante, nous allons montrer une conséquence de cette propriété.

5.3.2 Borne inférieure du compte réel

Dans cette partie on montre une conséquence immédiate de la propriété 2. En effet, si cette propriété est vérifiée, le nombre de groupes créés lors du partitionnement des données est une borne inférieure du compte réel.

COROLLAIRE 1

Si le paramètre ϵ utilisé dans l'algorithme de regroupement est tel que cet algorithme a la propriété de non duplication des groupes, alors le nombre m de groupes obtenus est une borne inférieure du nombre d'appareils présents.

Démonstration :

Soit $\mathcal{D}$ l'ensemble des données à partitionner et $(t_i)_{1\leq i\leq N}$ l'ensemble des téléphones dont au moins une RS est présente dans $\mathcal{D}$. Soit $(C_i)_{1\leq i\leq \hat{N}}$ le partitionnement obtenu après traitement par l'algorithme. Tous les C_i sont non vides.

La propriété 2 permet d'affirmer que pour tout k, il existe $i \in [\![1, \hat{N}]\!]$ tel que toutes les RS de t_k sont regroupées dans C_i. De plus, i est unique car les $(C_i)_{1\leq i\leq \hat{N}}$ sont disjoints. On note f la fonction qui à tout k associe i l'indice du groupe dans lequel toutes les RS de t_k sont regroupées.

f est une fonction de $[\![1, N]\!]$ dans $[\![1, \hat{N}]\!]$. Pour tout $i \in [\![1, \hat{N}]\!]$, C_i est non vide et il existe donc $k \in [\![1, N]\!]$ tel que C_i contient toutes les RS de t_k. C'est à dire :

$$\forall i \in [\![1, \hat{N}]\!], \exists k \in [\![1, N]\!], f(k) = i.$$

On en déduit que f est surjective. On en déduit donc que $card([\![1, \hat{N}]\!]) \leq card([\![1, N]\!])$ par le principe des tiroirs de Dirichlet, puis $\hat{N} \leq N$.

■

5.3.3 Erreur théorique de regroupement en fonction de l'erreur de la métrique

Dans cette partie nous montrons que sous certaines hypothèses, il est possible de relier théoriquement le nombre de groupes formés par l'algorithme de partitionnement 2 au nombre réel de téléphones et à une mesure de la précision de la métrique utilisée.

THÉORÈME

Soit une métrique d sur l'espace des Requêtes de Sonde admettant une distance critique ϵ^* et soit $\mathcal{A}$ l'algorithme de Vanhoef et al. renouvelé utilisant ϵ comme paramètre. On suppose que ce paramètre est suffisamment grand pour assurer le respect de la propriété 2.

Soient deux RS p_1 et p_2 provenant de téléphones différents $t_1 \neq t_2$. On note γ l'erreur de "confusion" : $\gamma = \mathbb{P}[d(p_1, p_2) < \epsilon | t_1 \neq t_2]$.

Enfin, notons N le nombre de téléphones dont au moins une RS est dans les données à partitionner.

Alors le nombre de groupes $\hat{N}$ formés par l'algorithme de regroupement $\mathcal{A}$ peut être modélisé par N transitions d'une chaîne de Markov à temps discret M_k telle que :

- $M_0 = 0$
- $M_{k+1} = M_k + X_k$, avec $X_k \sim \mathcal{B}((1-\gamma)^{M_k})$ où $\mathcal{B}(\theta)$ est la loi de probabilité d'une épreuve de Bernouilli de paramètre θ et k est le temps.

Démonstration : On note $(t_i)_{1 \leq i \leq N}$ l'ensemble des téléphones dont au moins une RS est dans l'ensemble à partitionner $\mathcal{D}$. L'ordre de ces téléphones est leur ordre d'apparition dans les données à partitionner. On suppose ces téléphones indépendants. On note $(\mathcal{D}_k)_{0 \leq k \leq N}$ des sous-ensembles de données à regrouper tels que :

- $\mathcal{D}_0 = \emptyset$.
- $\mathcal{D}_k$ contient toutes les RS de l'ensemble $\mathcal{D}$ avant la première RS de t_{k+1}. Ce sous-ensemble contient donc au moins une RS de chacun des téléphones $t_1, t_2, ..., t_k$, mais aucune RS des téléphones t_{k+1} à t_N.

On note $\hat{N}_k$ le nombre de groupes formés par le partitionnement de $\mathcal{D}_k$ par $\mathcal{A}$.

On a tout de suite $\hat{N}_0 = 0$. Nous cherchons à exprimer $\hat{N}_{k+1}$ en fonction de $\hat{N}_k$. Positionnons-nous à l'étape de l'algorithme de Vanhoef où $\mathcal{D}_k$ a déjà été partitionné en $\hat{N}_k$ groupes différents et la première RS de t_{k+1} est traitée. On note p cette RS.

On note $C_1, ..., C_{\hat{N}_k}$ les groupes déjà formés et $p_1, ..., p_{\hat{N}_k}$ leur dernière RS respective.

Il n'y a que deux cas possibles :

- Cas 1 : $\exists i \in [\![1, \hat{N}_k]\!]$ tel que $d(p_i, p) \leq \epsilon$. Dans ce cas, le nombre de groupes n'aug-

mente pas. Par la propriété 2, les RS $(p_i)_{1\leq i\leq \hat{N}_k}$ proviennent de téléphones différents et sont donc indépendantes. Par indépendance, la probabilité que le nombre de groupes n'augmente pas est donc :

$$\mathbb{P}[\exists i, d(p_i,p)\leq\epsilon] = 1-\mathbb{P}[\forall i, d(p_i,p)>\epsilon] = 1-\prod_{i=1}^{\hat{N}_k}\mathbb{P}[d(p_i,p)>\epsilon] = 1-(1-\gamma)^{\hat{N}_k}.$$

— Cas 2 : $\forall i\in[\![1,\hat{N}_k]\!], d(p_i,p)>\epsilon$. Dans ce cas un nouveau groupe est créé. Encore une fois par l'indépendance des RS $(p_i)_{1\leq i\leq \hat{N}_k}$, la probabilité que ce cas se produise est :

$$\mathbb{P}[\forall i, d(p_i,p)>\epsilon] = \prod_{i=1}^{\hat{N}_k}\mathbb{P}[d(p_i,p)>\epsilon] = (1-\gamma)^{\hat{N}_k}.$$

Dans la suite de l'algorithme, les autres RS de $\mathcal{D}_{k+1}\backslash\mathcal{D}_k$ ne créent pas de nouveau groupe par la propriété 2.

On a donc :

— $\hat{N}_{k+1}=\hat{N}_k$ avec probabilité $1-(1-\gamma)^{\hat{N}_k}$.

— $\hat{N}_{k+1}=\hat{N}_k+1$ avec probabilité $(1-\gamma)^{\hat{N}_k}$.

$\hat{N}_k$ est la chaîne de Markov cherchée.

■

5.3.4 Probabilité d'obtenir le compte $\hat{N}$ sachant N pour l'algorithme de Vanhoef et. al renouvelé

Dans cette partie, on utilise la chaîne de Markov précédemment défini pour calculer $\mathbb{P}[\hat{N}|N]$ où $\hat{N}$ est le nombre de groupes formés par l'algorithme de partitionnement et N le nombre de téléphones dans les données à partitionner.

On pose P la matrice de transition de la chaîne de Markov M telle que :

— $M_0=0$.

— $M_{k+1}=M_k+X_k$, avec $X_k\sim\mathcal{B}((1-\gamma)^{M_k})$.

P est une matrice carrée de $\mathcal{M}_{N+1}(\mathbb{R})$, triangulaire inférieure avec :

— $\forall i\in[\![0,N]\!], P_{i,i}=1-(1-\gamma)^i$.

— $\forall i\in[\![0,N-1]\!], P_{i+1,i}=(1-\gamma)^i$.

— $P_{i,j}=0$ partout ailleurs.

On a donc par les propriétés des chaînes de Markov :

$$\mathbb{P}[\hat{N}|N] = P^N\times u.$$

Où u est simplement le vecteur colonne d'état initial de taille $N+1$, $u = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

On montre l'évolution de ces probabilités sur la Figure 5.2 pour $\gamma = 0.15$ et $\gamma = 0.05$ par exemple. Ce graphique montre bien que plus le nombre N de téléphones est grand, plus l'incertitude sur $\hat{N}$ est grande. On y voit aussi l'impact de la valeur de γ sur cette incertitude.

5.3.5 Estimateur du maximum de vraisemblance et intervalle de confiance

Dans cette partie on utilise un estimateur du maximum de vraisemblance (EMV) pour estimer N à partir de $\hat{N}$. Dans toute la suite, nous notons $\tilde{N}$ les estimateurs pour les différencier de $\hat{N}$ le nombre de groupes directement formés par les algorithmes de partitionnement.

Ainsi l'estimation de N à partir des résultats de partitionnement est : $\tilde{N} = \underset{N}{\text{argmax}}(\mathbb{P}[\hat{N}|N])$.

Notons qu'il ne s'agit pas ici d'un EMV classique. Usuellement, l'EMV est bâti sur plusieurs mesures et la vraisemblance à maximiser est $\prod_{k=1}^{n} \mathbb{P}[\hat{N}_k|N]$ où les $\hat{N}_k$ sont n mesures suivant la même loi. Cela peut être intéressant dans des cas d'études où il est connu que N ne varie pas. Cependant, si l'objectif est de mesurer les variations de N en temps réel et à court terme, il semble moins pertinent d'utiliser plusieurs mesures pour calculer l'EMV.

Nous voulons également trouver un intervalle de confiance au seuil α pour N. Puisque $\tilde{N}$ n'utilise qu'une mesure, il n'est pas possible d'utiliser la normalité asymptotique de l'EMV. On cherche donc directement :

— k_1 tel que $\mathbb{P}[N > \tilde{N} + k_1|\hat{N}] < \dfrac{1-\alpha}{2}$.

— k_2 tel que $\mathbb{P}[N < \tilde{N} - k_2|\hat{N}] < \dfrac{1-\alpha}{2}$.

Notons que nous choisissons ici un intervalle de confiance asymétrique car les $\mathbb{P}[N|\hat{N}]$ ne sont pas symétriques autour de $\tilde{N}$. Remarquons tout d'abord que M, le nombre de RS à regrouper, est une borne supérieure du nombre de téléphones. On peut réécrire les probabilités cherchées en utilisant la formule de Bayes :

$$\mathbb{P}[N > \tilde{N} + k_1|\hat{N}] = \sum_{n=\tilde{N}+k_1+1}^{M} \mathbb{P}[N = n|\hat{N}] = \sum_{n=\tilde{N}+k_1+1}^{M} \frac{\mathbb{P}[\hat{N}|N = n] \times \mathbb{P}[N = n]}{\mathbb{P}[\hat{N}]}.$$

Ensuite, développons le terme $\mathbb{P}[\hat{\mathbb{N}}]$ en utilisant la formule des probabilités totales :

$$\mathbb{P}[N > \tilde{N} + k_1 | \hat{N}] = \sum_{n=\tilde{N}+k_1+1}^{M} \frac{\mathbb{P}[\hat{N}|N=n] \times \mathbb{P}[N=n]}{\sum_{m=0}^{M} \mathbb{P}[\hat{N}|N=m] \times \mathbb{P}[N=m]}.$$

Ici deux approches sont possibles. Si le cas d'étude permet de connaître les $\mathbb{P}[N]$, on peut terminer les calculs et trouver la valeur de k_1 avec cette formule. Cependant, à défaut, nous supposons l'équiprobabilité pour N et simplifions un peu la formule :

$$\mathbb{P}[N > \tilde{N} + k_1 | \hat{N}] = \sum_{n=\tilde{N}+k_1+1}^{M} \frac{\mathbb{P}[\hat{N}|N=n]}{\sum_{m=0}^{M} \mathbb{P}[\hat{N}|N=m]}.$$

Nous disposons des probabilités $\mathbb{P}[\hat{N}|N]$ grâce aux parties précédentes. Avec le même raisonnement sur k_2, on obtient :

$$\mathbb{P}[N < \tilde{N} - k_2 | \hat{N}] = \sum_{n=0}^{\tilde{N}-k_2-1} \frac{\mathbb{P}[\hat{N}|N=n] \times \mathbb{P}[N=n]}{\sum_{m=0}^{M} \mathbb{P}[\hat{N}|N=m] \times \mathbb{P}[N=m]}.$$

De même si l'on suppose l'équiprobabilité de N :

$$\mathbb{P}[N < \tilde{N} - k_2 | \hat{N}] = \sum_{n=0}^{\tilde{N}-k_2-1} \frac{\mathbb{P}[\hat{N}|N=n]}{\sum_{m=0}^{M} \mathbb{P}[\hat{N}|N=m]}.$$

On donne des exemples de ces intervalles de confiance au seuil $\alpha = 0.9$ sur la Figure 5.3. On voit sur cette figure qu'estimer les probabilités *a priori* $P[N]$ sur CRAWDAD permet de resserrer les intervalles de confiance. On voit également que la valeur de γ a un impact important sur la taille des intervalles de confiance. Ainsi, si l'on mesure $\hat{N} = 8$, l'estimation et les intervalles de confiance sont très différents selon la valeur de γ. Si $\gamma = 0.05$, on prédit $\tilde{N} = 9$ et un intervalle de confiance $[\![8, 16]\!]$. Pour $\gamma = 0.15$, on prédit $\tilde{N} = 15$ et un intervalle de confiance $[\![13, 30]\!]$.

Avant de conclure ce chapitre, nous allons calculer les complexités des deux algorithmes de partitionnement proposés.

5.3.6 Complexités des algorithmes de partitionnement en termes d'appels à la métrique

Nous calculons les complexités des algorithmes en termes du nombre d'appels à la fonction de distance. Pour cela, on note M le nombre de Requêtes de Sonde et N le nombre de téléphones.

Le regroupement par DBSCAN utilise invariablement la distance deux à deux entre toutes les RS. Par symétrie de la métrique, le nombre total d'appels à la fonction de distance est : $\frac{M \times (M-1)}{2}$.
Le regroupement par Vanhoef et. al renouvelé ne calcule qu'une distance par groupe déjà formé. On peut donc déjà affirmer que cet algorithme totalise moins d'appels à la fonction de distance. Calculons tout de même une borne supérieure simple du nombre d'appels à la métrique.

On note $\hat{N}_k$ le nombre de groupes formés par l'algorithme 2 après le traitement de la k-ième RS. Le nombre total d'appels à la fonction de distance est donc : $\sum_{k=1}^{M-1} \hat{N}_k \leq (M-1) \times \hat{N}$. (Avec $\hat{N} = N$ dans le pire des cas).

On en déduit que l'algorithme de Vanhoef et. al est plus rapide que DBSCAN.

5.4 Conclusion sur les algorithmes de partitionnement

Dans ce chapitre, nous avons proposé deux algorithmes pour le regroupement des Requêtes de Sonde. Ces algorithmes reposent sur la métrique définie dans le chapitre précédent. Nous avons montré que ces deux algorithmes ont la propriété intéressante de toujours donner une borne inférieure du décompte de téléphones.

Ensuite, nous avons mis en place une modélisation du nombre de groupes formés par l'algorithme de partitionnement de Vanhoef et. al renouvelé. Cette modélisation nous a permis de mettre au point un estimateur du maximum de vraisemblance $\tilde{N}$. Ensuite, nous avons établi un intervalle de confiance au seuil α pour cet estimateur. Enfin, nous avons montré que l'algorithme 2 est plus rapide que l'algorithme DBSCAN.

Dans le chapitre suivant, nous allons confronter notre métrique et nos algorithmes de partitionnement aux données de CRAWDAD, puis aux données que nous avons collecté manuellement afin de vérifier qu'elles n'invalident pas complètement le modèle. Enfin, nous allons apporter la conclusion de ce mémoire.

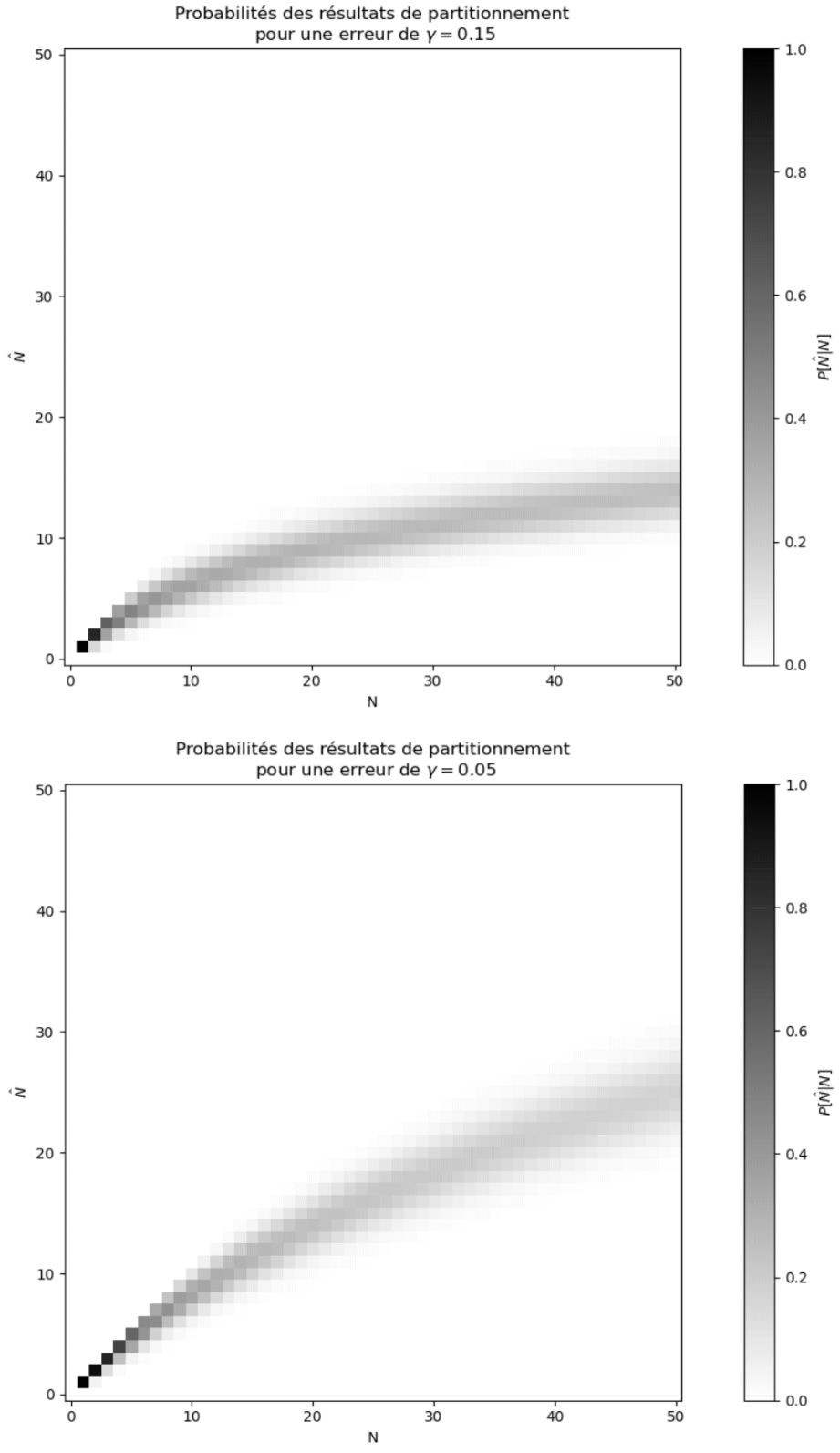

Figure 5.2 Probabilité $\mathbb{P}[\hat{N}|N]$ pour $\gamma = 0.15$ et $\gamma = 0.05$

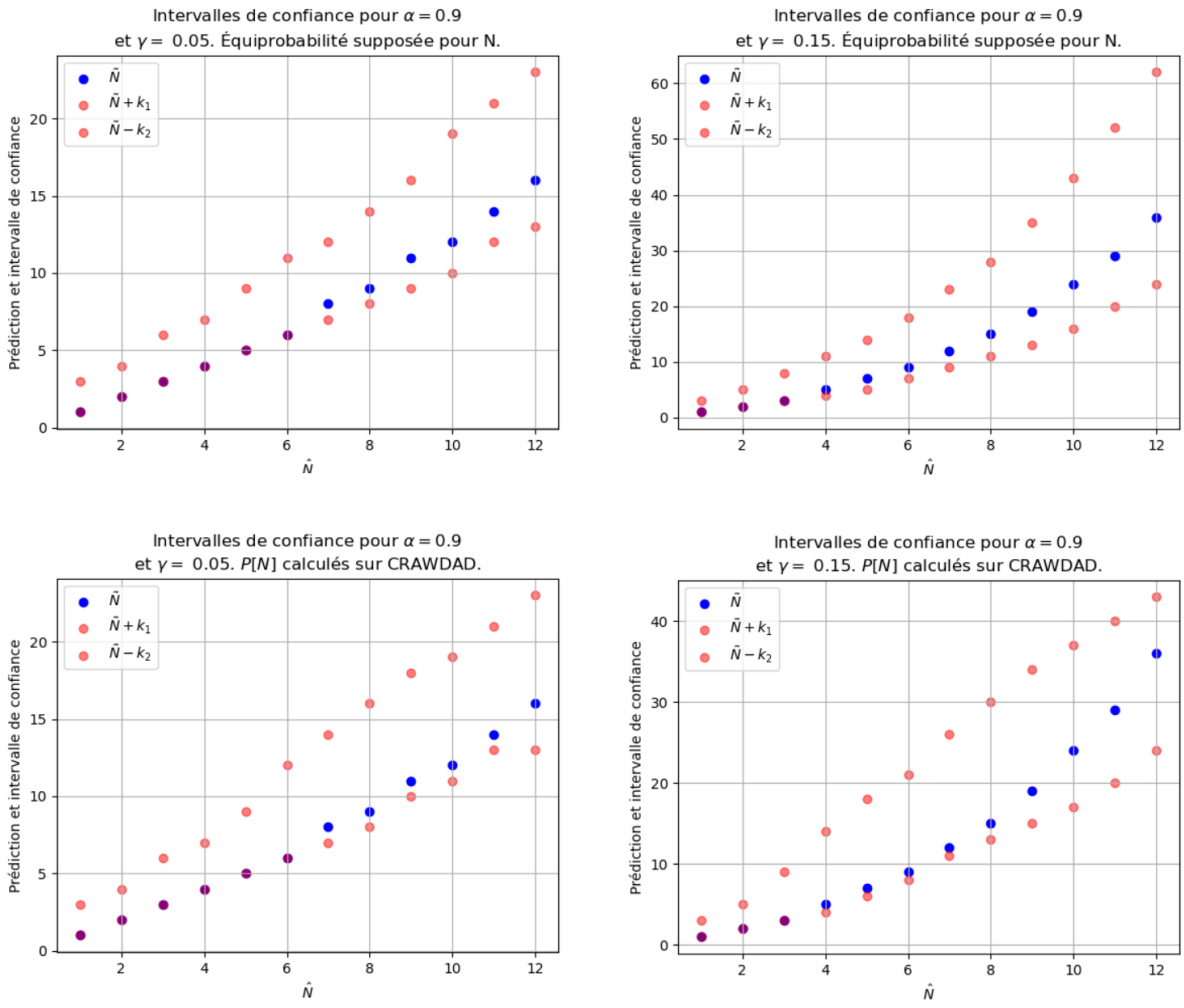

Figure 5.3 Intervalles de confiance au seuil $\alpha = 0.9$ pour la valeur de N en fonction de $\hat{N}$

CHAPITRE 6 APPLICATION ET ÉVALUATION DE LA MÉTHODE PROPOSÉE

Dans ce chapitre nous mettons en oeuvre l'intégralité de notre méthode sur les données de CRAWDAD. Nous entraînons les métriques définies au Chapitre 4, et les intégrons dans les algorithmes de partitionnement proposés au Chapitre 5. Nous corrigeons ensuite les estimations à l'aide des considérations mathématiques de la fin du Chapitre 5. Nous calculons l'erreur absolue moyenne en pourcentage et la comparons entre les algorithmes. Nous montrons que les objectifs de ce mémoire sont atteints avec le modèle théorique introduit. Enfin, nous utilisons les données de Requêtes de Sonde que nous avons collectées manuellement pour proposer un début de validation. Nous émettons cependant des réserves vis-à-vis de cette validation, qu'il faudrait pousser beaucoup plus loin avec plus de données annotées actuelles de Requêtes de Sonde. Cela nous permettra, au prochain chapitre, d'apporter la conclusion de ce mémoire.

6.1 Entraînement des métriques

Dans cette première section, nous entraînons les métriques proposées dans le Chapitre 4 sur un sous-ensemble des données de CRAWDAD. Nous utilisons ensuite un sous-ensemble différent afin de mesurer les propriétés de ces métriques. Tout d'abord, nous mesurons l'espérance de la fonction de perte, puis nous mesurons la distance critique ϵ^* et enfin nous mesurons la probabilité de confusion γ.

6.1.1 Convergence et perte des métriques

Dans cette partie, nous entraînons quatre versions différentes de la métrique et donnons leur perte dans le Tableau 6.1. Deux versions de ces métriques n'utilisent que les 10% meilleurs Éléments d'Information selon le critère défini au Chapitre 3. Les deux autres versions utilisent tous les Éléments d'Information. De même, deux versions utilisent les informations du RSSI et du temps d'arrivée, les deux autres versions ne les utilisent pas. Le Tableau 6.1 montre, comme attendu, que plus on prend en compte de champs des Requêtes de Sonde, plus l'espérance de la fonction de perte est basse pour la métrique. Ainsi notre meilleure métrique au sens de la fonction de perte, la métrique 4, atteint une perte moyenne de seulement 0.025.

Tableau 6.1 Liste des métriques et espérance de la fonction de perte

Numéro de la métrique	Éléments d'Information utilisés	RSSI et temps d'arrivée	Perte moyenne
1	Top 10%	Non	0.14
2	Top 10%	Oui	0.13
3	Tous	Non	0.035
4	Tous	Oui	0.025

6.1.2 Distance critique ϵ^* pour les métriques

Dans cette partie, nous vérifions que notre métrique admet bien une distance critique ϵ^* telle que définie dans le Chapitre 4.

La figure 6.1 donne la répartition des distances pour des paires aléatoires de RS de CRAWDAD provenant soit du même téléphone (en bleu) soit de téléphones différents (en rouge). Idéalement, on aimerait que la courbe bleue soit confinée le plus proche possible de l'axe des ordonnées (ϵ^* faible) et que l'aire sous la courbe rouge intersecte le moins possible l'aire sous la courbe bleue (γ faible). Cette figure montre que, en particulier pour les métriques 3 et 4, les RS en provenance des mêmes téléphones ont systématiquement une distance très faible. Cela illustre bien la propriété 2 d'existence d'une distance critique ϵ^* telle que $p_1, p_2, t_1 = t_2 \implies d(p_1, p_2) < \epsilon^*$.

Nous donnons les valeurs de ϵ^* pour chaque métrique dans le tableau 6.2. On voit que les métriques 3 et 4 ont une valeur de ϵ^* prometteuse, avec 0.13 et 0.22 respectivement. Pour les métriques 1 et 2, ϵ^* est malheureusement plutôt grand avec des valeurs supérieures à 1. Même si ces valeurs sont dues à des cas rares et isolés, cela remet en cause les parties de l'approche qui reposent sur ϵ^* pour les métriques 1 et 2.

Tableau 6.2 Liste des distances critiques ϵ^* pour chaque métrique

Numéro de la métrique	ϵ^*
1	1.4
2	3.6
3	0.13
4	0.22

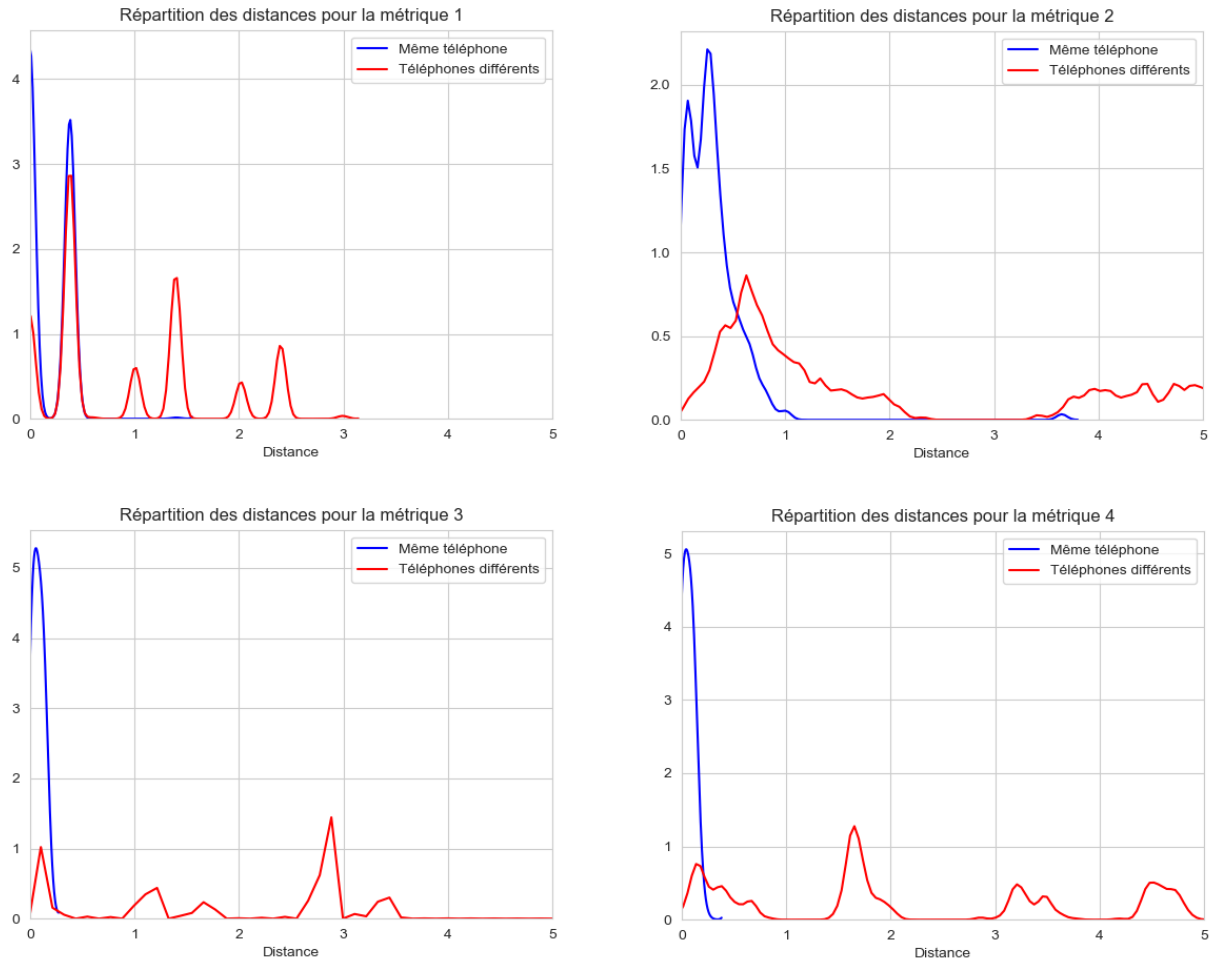

Figure 6.1 Répartitions des distances pour des paires de Requêtes de Sonde venant des mêmes téléphones et de téléphones différents

6.1.3 Valeurs de γ pour les métriques

La dernière caractéristique à mesurer pour chaque métrique est la probabilité de confusion γ, définie au Chapitre 5. La probabilité γ caractérise la tendance qu'a la métrique à associer deux RS provenant pourtant de téléphones différents. On donne une mesure de γ pour chacune des métriques dans le tableau 6.3. On voit que c'est la métrique 4 qui a le meilleur γ avec une valeur de 0.12. La métrique 3 suit de près avec 0.14. Les autres métriques ont un γ de plus de 0.4.

6.1.4 Sélection de la meilleure métrique

Étant données les mesures faites dans cette partie, tout semble indiquer que la métrique 4 est celle à retenir. En effet, même si elle a un ϵ^* plus grand que la métrique 3, c'est la métrique

Tableau 6.3 Liste des probabilités de confusion γ pour chaque métrique

Numéro de la métrique	γ
1	0.55
2	0.42
3	0.14
4	0.12

4 qui a le plus faible γ. La valeur de ϵ^* importe peu. C'est l'existence de ϵ^* qui importe. Sans surprise, c'est également la métrique avec la plus petite perte.

Dans toute la suite de ce chapitre, c'est donc la métrique 4 qui sera employée lors de la mise en oeuvre des algorithmes de partitionnement et de l'évaluation des performances de la méthode.

6.2 Évaluation du partitionnement

Dans cette partie nous testons les algorithmes de partitionnement DBSCAN et Vanhoef et. al renouvelé sur les données de CRAWDAD. Nous utilisons ensuite les nombres de groupes issus de ces partitionnements pour effectuer des prédictions du nombre de téléphones au regard des considérations mathématiques formulées à la fin du Chapitre 5. Nous calculons l'erreur relative dans chacun des cas pour comparer ces algorithmes. Nous mesurons également les temps d'exécution moyens de ces algorithmes et vérifions qu'ils permettent une estimation du nombre de téléphones en temps réel. Enfin, nous confrontons notre approche à un ensemble de validation limité que nous avons collecté nous-même avec 5 téléphones à notre disposition.

6.2.1 Regroupement avec Vanhoef et. al renouvelé

On donne dans cette partie les résultats du regroupement des RS avec l'algorithme de Vanhoef et. al renouvelé (Algorithme 2), la métrique 4 et $\epsilon = \epsilon^* = 0.22$.

La Figure 6.2 montre les résultats directs du regroupement, c'est-à- dire $\hat{N}$ en fonction de N. La Figure 6.3 montre ces mêmes résultats de regroupement superposés avec les probabilités théoriques calculées à l'aide de notre théorème. De la transparence est utilisée afin de visualiser les points superposés. La forte correspondance entre les points et les probabilités théoriques semble valider notre modélisation en chaîne de Markov.

Ces résultats préliminaires sont ensuite utilisés pour calculer l'estimateur du maximum de

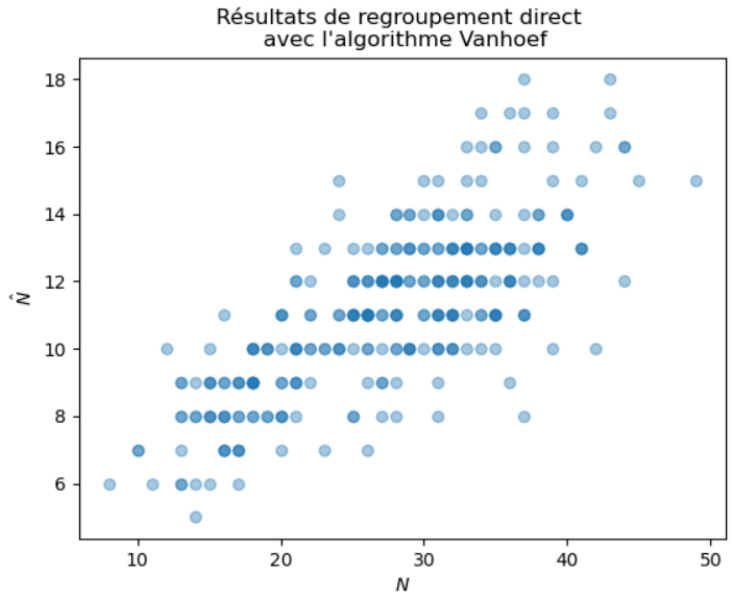

Figure 6.2 Résultats directs de regroupement sur CRAWDAD pour l'algorithme de Vanhoef et. al renouvelé

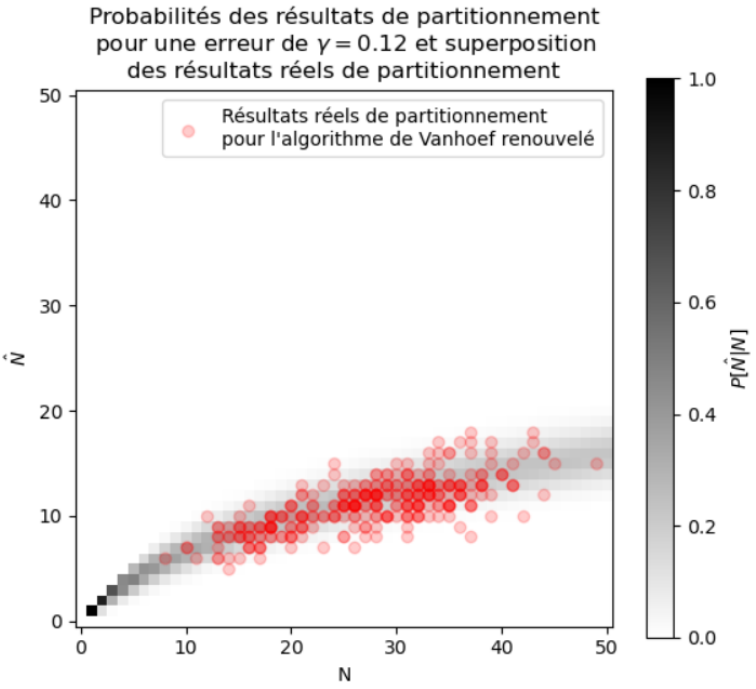

Figure 6.3 Résultats directs de regroupement sur CRAWDAD pour l'algorithme de Vanhoef et. al renouvelé superposé avec les probabilités théoriques

vraisemblance $\tilde{N}$. Cet estimateur ne requiert aucune hypothèse sur la distribution de N. On confronte cet estimateur au véritable nombre de téléphones N dans la Figure 6.4. Sur

cette figure, on trace en rouge la bissectrice des axes. L'erreur commise est alors simple à visualiser : c'est la distance horizontale entre chaque point de prédiction et la bissectrice des axes.

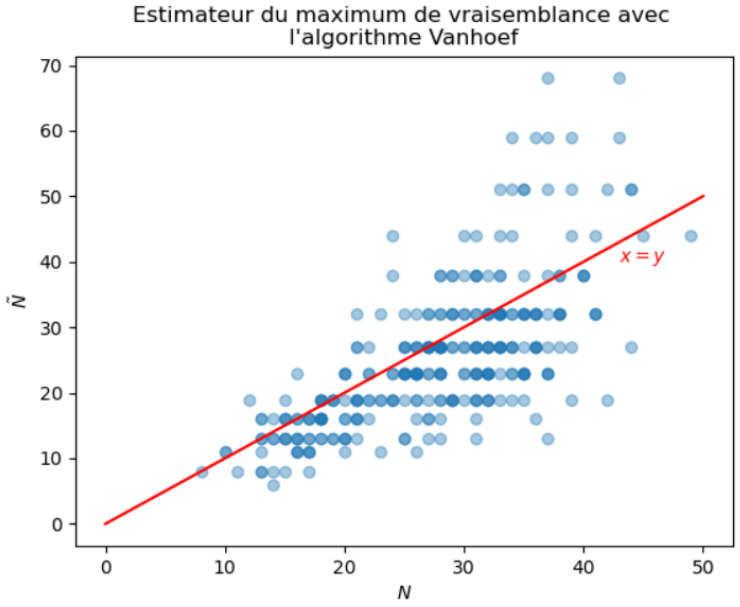

Figure 6.4 Estimateur du maximum de vraisemblance sur CRAWDAD pour l'algorithme de Vanhoef et. al renouvelé

Si l'on connaît la distribution *a priori* de N, on peut aussi construire l'estimateur non biaisé $\mathbb{E}[N|\hat{N}]$. Si l'on suppose que les N sont équiprobables, on obtient un résultat très similaire à celui de l'estimateur du maximum de vraisemblance. On donne les résultats de partitionnement avec cet estimateur en Figure 6.5. Encore une fois, l'erreur est simple à visualiser en utilisant la bissectrice des axes.

6.2.2 Regroupement avec DBSCAN

On donne dans cette partie les résultats du regroupement des RS avec DBSCAN, la métrique 4 et $\epsilon = \epsilon^* = 0.22$. La Figure 6.6 montre les résultats directs du regroupement, c'est-à-dire $\hat{N}$ en fonction de N. On voit que la relation entre $\hat{N}$ et N semble affine.

Nous n'avons pas démontré que la modélisation en chaîne de Markov était applicable aux résultats de partitionnement de l'algorithme DBSCAN. Cependant, comme mentionné au Chapitre 5, nous utilisons un cas particulier de l'algorithme DBSCAN pour nos essais. En effet, pour le paramètre $min\_pts = 1$, cette méthode de partitionnement est équivalente à une autre méthode appelée "partitionnement hiérarchique".

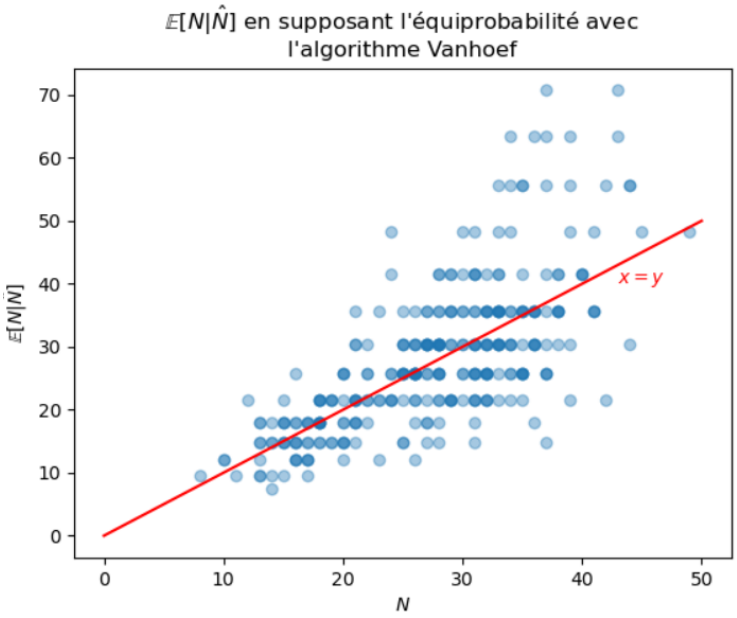

Figure 6.5 Estimateur $\mathbb{E}[N|\hat{N}]$ sur CRAWDAD pour l'algorithme de Vanhoef et. al renouvelé

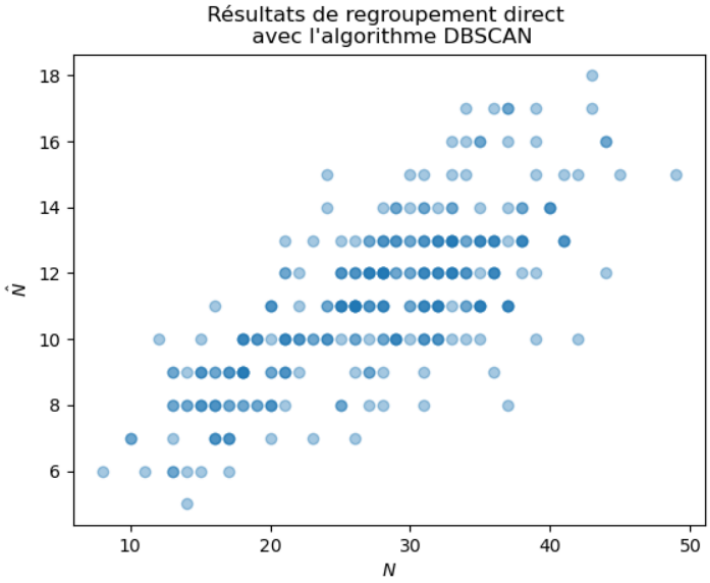

Figure 6.6 Résultats directs de regroupement sur CRAWDAD pour l'algorithme DBSCAN

Les résultats de partitionnement direct étant très similaires à ceux de l'algorithme de Vanhoef renouvelé, il semble raisonnable de penser que la modélisation en chaîne de Markov est également valable pour le partitionnement hiérarchique. Nous donnons donc les mêmes figures que celles de la section précédente pour l'algorithme DBSCAN.

La Figure 6.7 donne la superposition des résultats directs de partitionnement par DBSCAN

avec les probabilités théoriques calculées en utilisant notre théorème. Encore une fois, les données de partitionnement réel correspondent très bien à la modélisation en chaîne de Markov.

On donne enfin les résultats de l'estimateur du maximum de vraisemblance et de $\mathbb{E}[N|\hat{N}]$ pour DBSCAN dans les Figures 6.8 et 6.9.

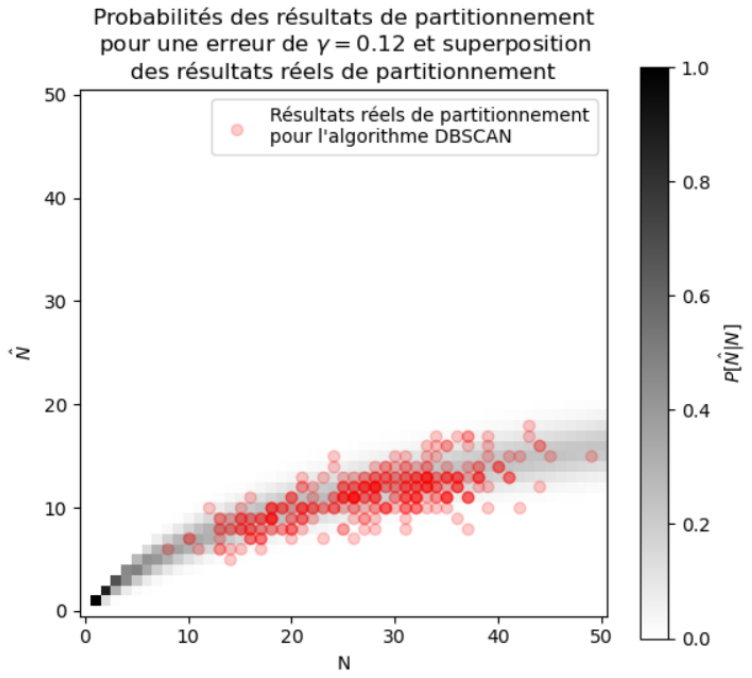

Figure 6.7 Résultats directs de regroupement sur CRAWDAD pour l'algorithme DBSCAN superposé avec les probabilités théoriques

6.2.3 Comparaison des algorithmes de partitionnement en précision et en temps

Dans cette partie, nous comparons les résultats des deux algorithmes de partitionnement pour chaque estimateur. Le critère de comparaison est l'Erreur Absolue Moyenne en Pourcentage (EAMP), définie comme :

$$EAMP = \frac{1}{n}\sum_{k=1}^{n}\left|\frac{\tilde{N}_k - N_k}{N_k}\right|.$$

où les $\tilde{N}_k$ sont les prédictions et les N_k sont les véritables valeurs du nombre de téléphones pour les n éléments de l'ensemble de test.

Le tableau 6.4 donne les valeurs de l'EAMP pour chacun des estimateurs et pour les deux

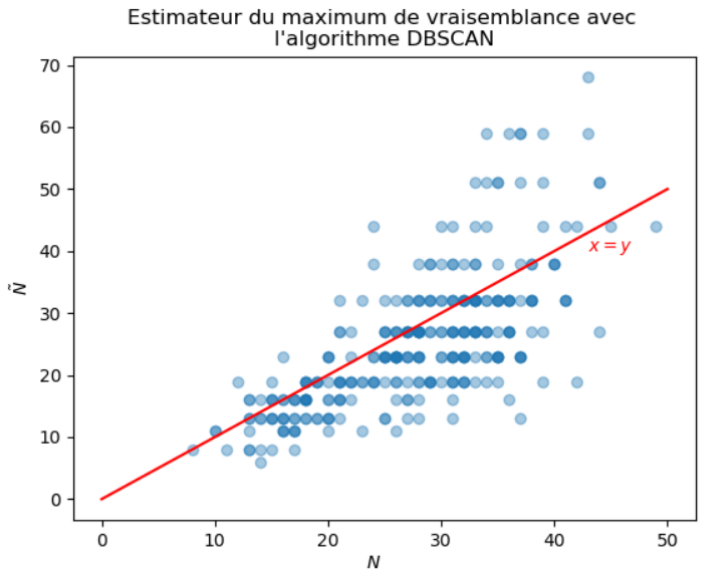

Figure 6.8 Estimateur du maximum de vraisemblance sur CRAWDAD pour l'algorithme DBSCAN

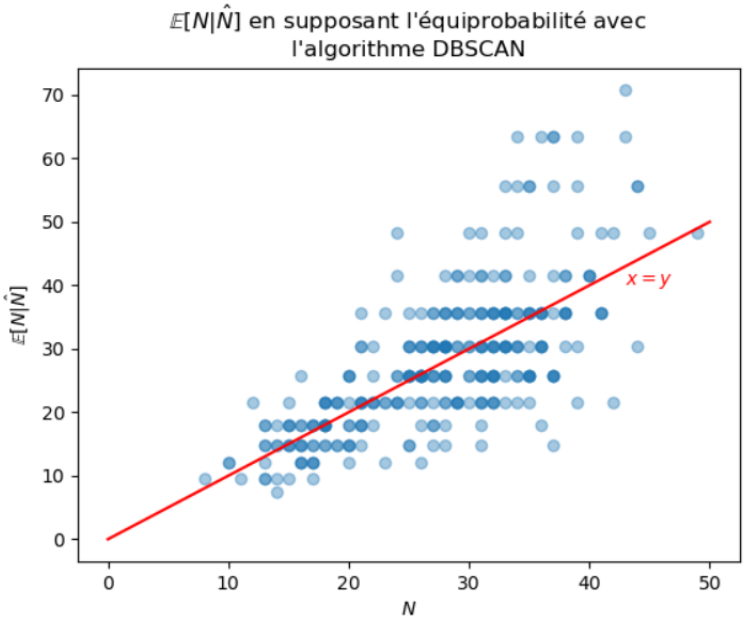

Figure 6.9 Estimateur $\mathbb{E}[N|\hat{N}]$ sur CRAWDAD pour l'algorithme DBSCAN

algorithmes. Les résultats sont les mêmes jusqu'au deuxième chiffre significatif, même si DBSCAN semble obtenir une précision légèrement supérieure. Cela ne permet pas d'affirmer qu'un algorithme est meilleur que l'autre. De même pour les estimateurs, la précision de l'estimateur du maximum de vraisemblance et de l'estimateur $\mathbb{E}[N|\hat{N}]$ sont les mêmes à 1%

près.

Tableau 6.4 Erreur moyenne relative sur CRAWDAD pour chaque estimateur et algorithme de partitionnement

Algorithme	Estimateur du maximum de vraisemblance $\tilde{N}$	Estimateur de la moyenne $\mathbb{E}[N\|\hat{N}]$
Vanhoef et. al renouvelé	21.9%	21.3%
DBSCAN (partitionnement hiérarchique)	21.8%	21.0%

Pour choisir l'algorithme à utiliser, nous proposons donc de considérer leur complexité respective et de ne retenir que le plus rapide. Ce travail a été fait à la fin du Chapitre 5. L'algorithme le plus rapide est celui de Vanhoef et. al renouvelé. C'est donc celui à retenir pour notre approche.

On donne en Tableau 6.5 les temps d'exécution moyens pour chaque algorithme, en secondes. Le nombre de RS à regrouper est choisi égal à 100. On donne également le temps moyen ΔT entre la première et la dernière RS reçue pour ces intervalles. Cela confirme que l'algorithme de Vanhoef et. al renouvelé est plus rapide que DBSCAN. On remarque néanmoins que les deux algorithmes parviennent à effectuer le regroupement en un temps environ cinq fois inférieur au temps d'arrivée de 100 RS. Ils fonctionnent donc largement en temps réel.

Tableau 6.5 Temps moyen d'exécution des algorithmes et temps moyen d'arrivée de 100 Requêtes de Sonde

Temps moyen pour DBSCAN	Temps moyen pour Vanhoef et. al renouvelé	ΔT moyen
6.3 s	5.2 s	31.1 s

6.3 Validation

Dans cette partie, nous appliquons notre méthode à un ensemble de données collectées manuellement sur 5 téléphones à notre disposition. Le résultat de partitionnement direct donne $\hat{N} = 3$ pour les deux algorithmes. Selon notre modélisation en chaîne de Markov, l'intervalle de confiance au seuil $\alpha = 0.9$ pour N est $[\![3, 6]\!]$ et contient bien le nombre N réel de téléphones.

Même si ce résultat est rassurant, on ne peut malheureusement pas en déduire que notre méthode fonctionne sans ajustement dans l'état actuel du protocole 802.11. Il faudrait pour cela effectuer des essais sur un nombre bien plus grand de téléphones. Notamment, il faudrait vérifier les propriétés des métriques, et recalculer la valeur de ϵ^* et de γ. Idéalement, il faudrait même directement entraîner les métriques sur un ensemble de données annotées plus récent que CRAWDAD.

Nous avons désormais tous les éléments nécessaires pour construire une discussion autour des résultats de notre travail, puis apporter la conclusion de ce mémoire.

CHAPITRE 7 CONCLUSION ET RECOMMANDATIONS

Ce chapitre présente une discussion des résultats de notre méthode et apporte la conclusion de ce mémoire.

7.1 Discussion

Les résultats de notre approche sont encourageants. En effet, sans utiliser l'adresse MAC, nous parvenons à atteindre une erreur de seulement 21% sur les données de CRAWDAD en utilisant DBSCAN. Cela constitue une amélioration de l'approche de Uras et al. (2020) qui utilise le même algorithme de partitionnement et atteint une précision de 65.2%. De plus, contrairement aux articles de l'état de l'art, notre méthode introduit des intervalles de confiance sur la prédiction du nombre d'appareils.

Il existe des cas d'études pour lesquels l'erreur moyenne est satisfaisante. Par exemple, lorsque le décompte est utilisé à des fins statistiques pour déterminer des tendances de déplacement dans une ville. Cependant, si le compte est utilisé en temps réel, une erreur moyenne de 21% peut se révéler trop importante. Par exemple, s'il est important de faire la distinction entre de très faibles nombres de personnes (un, deux ou trois).

Il est toutefois intéressant de remarquer que l'erreur d'estimation n'est mesurée que dans le cas où une seule mesure est effectuée pour le calcul de l'EMV. Dans les cas d'étude où la vraisemblance est calculée à l'aide de plusieurs mesures, nous pouvons nous attendre à une précision améliorée en raison de la normalité asymptotique de l'EMV.

Il est cependant important de noter que notre approche reste largement théorique. Notamment, sa précision et ses intervalles de confiance sont particulièrement dépendants de la valeur de γ, comme l'illustre la Figure 5.3. Même si cette valeur calculée sur CRAWDAD est suffisamment faible pour assurer de bons résultats pour notre approche, rien ne garantit que cette grandeur n'augmente pas dans le contexte actuel du protocole WiFi. En effet, notre validation est limitée, et idéalement il faudrait disposer de données annotées similaires à CRAWDAD mais plus récentes pour calculer notre métrique.

De plus, le lien entre le nombre d'appareils détectés et le nombre de personnes présentes reste à établir. Ce lien peut par exemple prendre la forme d'un facteur multiplicatif, traduisant la probabilité que chaque personne possède avec elle un téléphone dont le WiFi est activé.

Enfin, il est intéressant de remarquer que notre métrique est un cas particulier d'une distance appelée distance de Mahalanobis. Utiliser directement la distance de Mahalanobis pourrait

permettre d'améliorer la métrique en prenant en compte les interactions entre les différents champs, deux à deux. Les avantages d'utiliser une distance de Mahalanobis sont à étudier en tenant compte de son principal désavantage sur notre approche : le nombre important de paramètres introduits.

Nous pouvons désormais apporter la conclusion de ce mémoire.

7.2 Conclusion

Notre objectif était de créer une méthode qui, lorsque mise en oeuvre dans un système embarqué indépendant en énergie, permet d'estimer le nombre de personnes présentes aux alentours d'un point donné en temps réel. En raison des contraintes énergétiques et étant donné l'état de l'art, nous avons opté pour une approche exploitant le protocole 802.11 de connexion WiFi. Afin de répondre à cette problématique, nous l'avons d'abord subdivisée en trois sous-objectifs.

Le premier sous-objectif était l'élaboration d'une nouvelle métrique permettant de comparer deux Requêtes de Sonde pour mesurer leur similarité. Nous avons atteint cet objectif en définissant une métrique simple, inspirée de la distance de Hamming et prenant en compte à la fois les données catégoriques des RS et leurs données scalaires. Nous avons ensuite optimisé les coefficients de cette métrique en utilisant une descente de gradient stochastique et une fonction de perte astucieuse. Nous avons montré que la métrique obtenue avait des propriétés notables, telle que l'existence d'une distance critique ϵ^* assurant que les RS provenant des mêmes téléphones soient évaluées comme étant à une distance inférieure à ϵ^*.

Le second sous-objectif était l'amélioration de méthodes existantes pour le regroupement des Requêtes de Sonde par téléphone d'origine. Le premier algorithme que nous avons retenu est l'algorithme de partitionnement DBSCAN, qui a fait ses preuves pour le regroupement de RS dans les travaux de Uras et al. (2020). Le second algorithme retenu est celui de Vanhoef et al. (2016). Nous avons modifié le fonctionnement de cet algorithme pour qu'il puisse inclure nos métriques. Nous avons ensuite montré qu'en travaillant de concert avec nos métriques, ces algorithmes obtiennent des propriétés utiles. Notamment, ils donnent une borne inférieure du compte de téléphones aux alentours.

Le dernier sous-objectif était l'analyse de l'erreur commise et l'établissement d'un intervalle de confiance sur le nombre d'appareils détectés. Nous avons montré que le nombre de groupes créés par l'algorithme de Vanhoef renouvelé utilisant nos métriques pouvait être modélisé par une chaîne de Markov. Nous avons utilisé cette modélisation afin de définir un estimateur du maximum de vraisemblance n'utilisant qu'une seule mesure $\tilde{N}$ ainsi qu'un intervalle de

confiance au seuil α pour le nombre de téléphones présents.

Nous avons enfin évalué notre approche sur les données de CRAWDAD. Avec ces données, l'erreur moyenne absolue en pourcentage commise lors de l'estimation est d'environ 21%. Cette erreur est inférieure à celle annoncée par Uras et al. (2020) dans leur approche utilisant DBSCAN (précision de 65.2%).

RÉFÉRENCES

M. V. Barbera, A. Epasto, A. Mei, S. Kosta, V. C. Perta, et J. Stefa, "CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10)", Downloaded from https://crawdad.org/sapienza/probe-requests/20130910, Sep. 2013. DOI : 10.15783/C76C7Z

M. Bouazizi et T. Ohtsuki, "An infrared array sensor-based method for localizing and counting people for health care and monitoring", 2020, pp. 4151 – 5. En ligne : http://dx.doi.org/10.1109/EMBC44109.2020.9176199

M. Braun, S. Krebs, F. B. Flohr, et D. M. Gavrila, "Eurocity persons : A novel benchmark for person detection in traffic scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019. DOI : 10.1109/TPAMI.2019.2897684

C. Chilipirea, C. Dobre, M. Baratchi, et M. van Steen, "Identifying movements in noisy crowd analytics data", 06 2018, pp. 161–166. DOI : 10.1109/MDM.2018.00033

H. Choi, T. Matsui, M. Fujimoto, et K. Yasumoto, "Simultaneous crowd counting and localization by wifi csi", dans *International Conference on Distributed Computing and Networking 2021*, série ICDCN '21. New York, NY, USA : Association for Computing Machinery, 2021, p. 239–240. DOI : 10.1145/3427796.3430000. En ligne : https://doi.org/10.1145/3427796.3430000

X. Chu, A. Zheng, X. Zhang, et J. Sun, "Detection in crowded scenes : One proposal, multiple predictions", *CoRR*, vol. abs/2003.09163, 2020. En ligne : https://arxiv.org/abs/2003.09163

D. Dai Zovi et S. Macaulay, "Attacking automatic wireless network selection", dans *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, 2005, pp. 365–372. DOI : 10.1109/IAW.2005.1495975

B. M. P. M. David A. Westcott, David D. Coleman, "Cwap certified wireless analysis professional official study guide : Exam pw0-270", 2011.

M. Ester, H.-P. Kriegel, J. Sander, et X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise". AAAI Press, 1996, pp. 226–231.

E. Fenske, D. Brown, J. Martin, T. Mayberry, P. Ryan, et E. C. Rye, "Three years later : A study of mac address randomization in mobile devices and when it succeeds", *Proceedings*

on Privacy Enhancing Technologies, vol. 2021, pp. 164 – 181, 2021.

J. Filipic, M. Biagini, I. Mas, C. D. Pose, J. I. Giribet, et D. R. Parisi, “People counting using visible and infrared images”, *Neurocomputing*, vol. 450, pp. 25 – 32, 2021.

D. Halperin, W. Hu, A. Sheth, et D. Wetherall, “Tool release : Gathering 802.11n traces with channel state information”, *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, jan 2011. DOI : 10.1145/1925861.1925870. En ligne : https://doi.org/10.1145/1925861.1925870

A. Hidayat, S. Terabe, et H. Yaginuma, “Estimating bus passenger volume based on a wi-fi scanner survey”, *Transportation Research Interdisciplinary Perspectives*, vol. 6, p. 100142, 2020. DOI : https://doi.org/10.1016/j.trip.2020.100142. En ligne : https://www.sciencedirect.com/science/article/pii/S2590198220300531

Y. Li, J. Barthelemy, S. Sun, P. Perez, et B. Moran, “A case study of wifi sniffing performance evaluation”, *IEEE Access*, vol. 8, pp. 129 224–129 235, 2020. DOI : 10.1109/ACCESS.2020.3008533

T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, et C. L. Zitnick, “Microsoft COCO : common objects in context”, *CoRR*, vol. abs/1405.0312, 2014. En ligne : http://arxiv.org/abs/1405.0312

E. Longo, A. E. C. Redondi, et M. Cesana, “Pairing wi-fi and bluetooth mac addresses through passive packet capture”, dans *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2018, pp. 1–4. DOI : 10.23919/MedHocNet.2018.8407082

J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, et D. Brown, “A study of MAC address randomization in mobile devices and when it fails”, *CoRR*, vol. abs/1703.02874, 2017. En ligne : http://arxiv.org/abs/1703.02874

Ministère des Transports, “Rapport de gestion 2020-2021 du ministère des transports au québec”, 2020-2021. En ligne : https://cdn-contenu.quebec.ca/cdn-contenu/adm/min/transports/publications-adm/rapport-annuel-de-gestion/RA_rapport_annuel_2020-2021_MTQ.pdf?1632844429

M. Mizutani, A. Uchiyama, T. Murakami, H. Abeysekera, et T. Higashino, “Towards people counting using wi-fi csi of mobile devices”, dans *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6. DOI : 10.1109/PerComWorkshops48775.2020.9156098

O. Oshiga, H. U. Suleiman, S. Thomas, P. Nzerem, L. Farouk, et S. Adeshina, “Human detection for crowd count estimation using csi of wifi signals”, dans *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, 2019, pp. 1–6. DOI : 10.1109/ICECCO48375.2019.9043195

C. Raghavachari, V. Aparna, S. Chithira, et V. Balasubramanian, “A comparative study of vision based human detection techniques in people counting applications”, *Procedia Computer Science*, vol. 58, pp. 461–469, 2015, second International Symposium on Computer Vision and the Internet (VisionNet’15). DOI : https://doi.org/10.1016/j.procs.2015.08.064. En ligne : https://www.sciencedirect.com/science/article/pii/S1877050915021754

J. Redmon, S. Divvala, R. Girshick, et A. Farhadi, “You only look once : Unified, real-time object detection”, 2016.

L. Schauer, M. Werner, et P. Marcus, “Estimating crowd densities and pedestrian flows using wi-fi and bluetooth”, 01 2014. DOI : 10.4108/icst.mobiquitous.2014.257870

E. Schubert, J. Sander, M. Ester, H. P. Kriegel, et X. Xu, “Dbscan revisited, revisited : Why and how you should (still) use dbscan”, *ACM Trans. Database Syst.*, vol. 42, no. 3, jul 2017. DOI : 10.1145/3068335. En ligne : https://doi.org/10.1145/3068335

S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, et J. Sun, “Crowdhuman : A benchmark for detecting human in a crowd”, *arXiv preprint arXiv :1805.00123*, 2018.

U. Singh, J.-F. Determe, F. Horlin, et P. D. Doncker, “Crowd forecasting based on wifi sensors and lstm neural networks”, *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6121–6131, 2020. DOI : 10.1109/TIM.2020.2969588

Statistique Canada, 2010. En ligne : https://www150.statcan.gc.ca/n1/pub/11-008-x/2011002/article/11531-fra.htm

M. Uras, R. Cossu, E. Ferrara, O. Bagdasar, A. Liotta, et L. Atzori, “Wifi probes sniffing : an artificial intelligence based approach for mac addresses de-randomization”, dans *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1–6. DOI : 10.1109/CAMAD50429.2020.9209257

M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, et F. Piessens, “Why mac address randomization is not enough : An analysis of wi-fi network discovery mechanisms”, dans *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, série ASIA CCS ’16. New York, NY, USA : Association for

Computing Machinery, 2016, p. 413–424. DOI : 10.1145/2897845.2897883. En ligne : https://doi.org/10.1145/2897845.2897883

Y. Zhou, B. P. L. Lau, Z. Koh, C. Yuen, et B. K. K. Ng, “Understanding crowd behaviors in a social event by passive wifi sensing and data mining”, *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4442–4454, 2020. DOI : 10.1109/JIOT.2020.2972062