

Raportare proiect Airdocs, Etapa 2

Laura Ruse (Dragomir), Costin Carabaș, Dragoș Niculescu

November 2021



Proiectul Airdocs a îndeplinit următoarele obiective în anul 2021:

- implementarea aplicației Android
- implementarea serverului Python
- colectarea și analiza datelor
- proiectarea metodelor de măsurare a similarităților între semnăturile colectate în clădiri
- publicarea codului client/server la adresa <https://github.com/airdocs>
- o publicație în jurnal Q2, IF 3.44

1 Introducere

AirDocs (Documents in the Air) este un sistem middleware care permite plasarea virtuală a documentelor și regăsirea lor la diverse locții din interiorul clădirilor fără a necesita un sistem de poziționare în interior. Are următoarele proprietăți: **1.** se bazează pe infrastructura existentă WiFi sau BLE existentă în multe instituții și locuințe **2.** necesită un singur server disponibil în intranet sau Internet, și o aplicație care poate fi instalată pe orice telefon smart **3.** nu necesită antrenament, sau implementarea unui serviciu de localizare **4.** permite multe aplicații care presupun plasarea naturală și regăsirea documentelor la locații fizice

Acest middleware va oferi trei primitive aplicațiilor (Figura 1):

S = CreateSgn() recoltează statistici de semnal specifice locației folosind dispozitivul mobil (WiFi, 3G, sunet, etc) și crează o semnătură multidimensională care poate fi deosebită de orice altă semnătură din clădire

Put(S, document) stochează pe server un document cu semnătura S. Semnătura este creată de un mobil, dar indexarea semnăturilor și stocarea se face pe un server din intranet (sau Internet).

Get(S) mobilul recoltează semnătura curentă S, și interoghează serverul pentru o listă de documente care au semnături asemănătoare. Aceste documente se presupune că au fost stocate la aceea locație. Serverul efectuează o căutare în spațiul semnăturilor, așadar locațiile geografice reale un sunt necesare și nu sunt stocate în baza de date (dacă nu sunt disponibile).

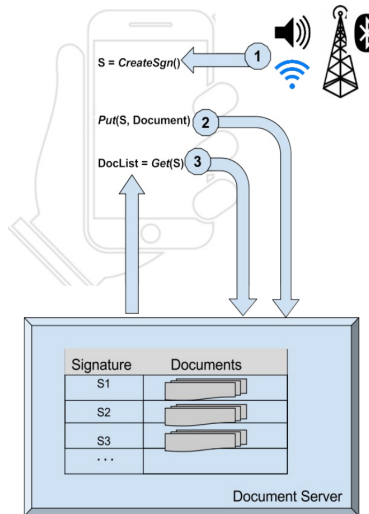


Figure 1: Primitivele Put si Get pe baza cărora sunt folosite semnăturile. Funcția Put plasează un document ”în aer” la o anumită locație în clădire identificată de semnătura S. Funcția Get regăsește documentele care au o semnătură asemănătoare cu S.

2 Implementarea clientului Android

Clientul AirDocs este folosit pentru a plasa și pentru a cauta documente într-o anumită locație. La plasarea și la căutarea unui document, cererea va fi însoțită de semnătura locației respective, care este generată de către middleware-ul AirDocs. Middleware-ul AirDocs include metode 1) pentru trimiterea documentelor către server, împreună cu semnătura asociată, 2) pentru căutarea documentelor de pe server care se potrivesc cu anumită semnătură recent colectată și 3) pentru ștergerea documentelor de pe server.

Semnătura va fi creată de middleware prin scanarea și obținerea informațiilor despre: rețelele WiFi, dispozitivele BLE, turnurile de telefonie mobilă, GPS și audio. Semnătura este de asemenea însoțită de alte meta informații precum: identificatorul dispozitivului, tipul dispozitivului și versiunea sistemului de operare.

Middleware-ul AirDocs folosește serviciile de sistem Android pentru a obține informațiile necesare pentru a construi semnătura:

- WifiManager - pentru a obține informații despre rețelele WiFi
- BluetoothLeScanner - pentru a obține informații despre dispozitivele BLE din acea locație
- TelephonyManager - pentru a obține informații despre turnurile de telefonie mobilă

- LocationManager - pentru a obține coordonatele GPS ale locației dacă sunt disponibile
- MediaRecorder - pentru a înregistra sunetul ambiental

Mesajele transmise între client și server sunt incluse în mesaje HTML de tip POST ce includ un JSON. Middleware-ul oferă suport pentru 5 tipuri diferite de mesaje: 1) TEST - mesaj de test, cu utilitate în etapa de dezvoltare, 2) POST - mesaj de plasare al unui document, 3) SEARCH - mesaj de căutare a unui document, 4) DELFILE - ștergerea unui document, 5) DELALL - ștergerea tuturor documentelor plasate de un anumit dispozitiv.

Mesajul de plasare al unui document include semnătura, meta-informațiile, fișierul efectiv, numele fișierului (incluzând extensia), tipul fișierului, descrierea fișierului. De asemenea, se poate plasa un text, fără un anumit fișier - în cazul acesta se va specifica în JSON tipul "text/plain". Răspunsul server-ului va fi un HTML cu mesajul "Successful Sending" în cazul salvării a documentului pe server, sau un mesaj de eroare în caz contrar.

Mesajul de căutare al unui document include semnătura, meta-informațiile și threshold-ul folosit pentru căutare. Răspunsul server-ului va fi un JSON ce va include fișierele găsite și informațiile acestora: identificatorul fișierului, similaritatea, numele fișierului, descrierea fișierului și conținutul fișierului serializat.

Interfața grafică a clientului include mai multe activități (ferestre), printre care: activitatea de plasare a unui fișier, activitatea de cautare a fișierelor, activitatea de configurare. Comunicarea între activități și serviciu se face prin Messenger și prin salvarea/restaurarea valorilor din SharedPreferences.

Activitatea de plasare a unui fișier poate primi un fișier printr-un Intent, în urma unui "Share" din altă aplicație. Aceasta acceptă un număr limitat de tipuri de fișiere (din motive de securitate): text/plain, image/*, audio/*, video/*, application/pdf, application/msword, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/zip. În cazul în care fișierul este o imagine, se va afișa un thumbnail în activitate. Utilizatorul poate completa descrierea și apoi poate trimite fișierul către server. În timpul acesta, serviciul generează semnătura și împachetează toate informațiile în JSON, apoi îl trimite la server.

Activitatea de căutare a fișierelor include un buton pentru pornirea căutării. Serviciul va genera semnătura, va împacheta informațiile în JSON și le va trimite la server. În cazul unei căutări de succes, va primi de la server JSON-ul cu informațiile despre fișierele găsite și va afișa o listă ce include: numele fișierului, descriere fișierului, un thumbnail (dacă este imagine). Fișierele sunt ordonate descrescător după similaritatea atașată în JSON. La apăsarea pe numele fișierului, se va deschide fișierul în aplicația corespunzătoare tipului respectiv de fișier.

Pentru fiecare fișier afișat în activitatea de căutare, există opțiunea de a fi șters de pe server. Serviciul va genera un JSON care va include identificatorul fișierului ce se dorește a fi șters. Un fișier poate fi șters doar de pe dispozitivul de pe care a fost plasat inițial (același utilizator). În caz de succes, fișierul va fi șters și din lista afișată în activitate.

Activitatea de configurare poate fi accesată din meniu și include următoarele opțiuni ce pot fi configurate: numărul de scanări, IP-ul și portul serverului, threshold-ul, componentele semnăturii (WiFi, BLE, cellular, GPS, Audio). Odată salvate aceste informații, ele vor fi stocate în SharedPreferences, de unde vor fi preluate de către activități și servicii. Tot din această activitate, se pot șterge toate documentele care au fost plasate de pe acest dispozitiv.

3 Implementarea serverului Python

3.1 Introducere

Serverul este responsabil cu stocarea documentelor și a semnăturilor asociate acestora, precum și cu identificarea documentului adecvat pentru o anumită semnătură. Atunci când un utilizator caută documentele dintr-o locație, serverul este interogată cu semnătura curentă a utilizatorului. În continuare, se compară semnătura colectată cu alte semnături stocate în baza de date, folosind o funcție de (di)similaritate. Se identifică cea mai asemănătoare semnătură, iar documentul asociat este preluat, transmis către client și afișat în aplicație. Dintre primitivile menționate în Introducere, `Put(S, document)` și `Get(S)` sunt implementate cu ajutorul serverului.

Pe server, documentele sunt indexate după semnături, pe baza asemănărilor dintre ele. Pentru o anumită interogare de semnătură, serverul poate (i). Prelua documentul cu cea mai mică diferență cu interogarea sau (ii). Preia toate documentele cu diferențe sub un prag. Deoarece pozițiile fizice reale nu sunt cunoscute, baza de date de semnături de pe server trebuie organizată folosind metode de grupare și etichetare.

Implementarea și testarea serverului AirDocs a fost făcută în două moduri: (i). expunerea serverului pe un port cunoscut, astfel încât aplicația pentru smartphone să poată descoperi cu ușurință instanțe ale serverelor cu documente și (ii). cu un server plasat în cloud, iar aplicația comunică cu serverul expus de noi fără să fie legată administrativ de un spațiu fizic.

3.2 Implementare

Serverul AirDocs comunică cu modulul smartphone AirDocs și este responsabil pentru două scenarii. În primul rând, primește un nou document împreună cu o semnătură asociată și le stochează într-o bază de date specifică clădirii, instituției sau grupului. În al doilea rând, răspunde la căutări bazate pe semnături, identificând documente cu disimilarități/deosebiri reduse. Cele mai asemănătoare semnături și documentele asociate acestora sunt trimise pe mobil și apoi afișate în aplicație.

Aplicația smartphone poate cere documente în două moduri posibile: (i) cel mai apropiat document sau (ii) documentele din apropierea mea. Pentru cel de-al doilea caz, am implementat o variabilă a (di)similarității cu valori cuprinse

între 0 și 1, unde 0 este locația cu semnătura identică și 1 sunt locațiile depărtate (la peste 15 metri).

Cu o implementare naivă a bazei de date liniare, inserările ar lua $O(1)$ și căutările $O(n)$, unde n este numărul de documente stocate pe server. AirDocs stochează o bază de date spațială, în sensul că documentele sunt legate de locație, dar locațiile reale ale documentelor nu sunt cunoscute. Prin urmare, majoritatea metodelor de indexare a informațiilor spațiale, cum ar fi *M-trees* și *R*-trees*, utilizate pentru optimizarea bazelor de date cu amprente digitale etichetate cu poziție nu sunt aplicabile.

Cu toate acestea, diferențele dintre documente impun o anumită structură în spațiul semnalului. O grupare/clasificare spațială naturală ar trebui să producă o structură ierarhică în clădiri, etaje, încăperi. Acest lucru ar permite căutarea într-o structură de tip arbore care utilizează relații spațiale de incluziune și adiacență.

Pentru clasificare, folosim un sistem multi-etichetare, unde clusterurile sunt formate în funcție de cel mai puternic AP observat, ceea ce definește aria procesului de căutare. Când primește un document, serverul atribuie etichete q , unde q este numărul celor mai puternice AP-uri vizibile în amprenta WiFi. Pentru fiecare interogare, serverul caută apoi numai în liste q asociate cu cele mai puternice AP-uri. Din cauza instabilităților WiFi de interior, cel mai puternic AP ar putea să nu fie întotdeauna același și întrebarea este **cum determinăm valoarea minimă a q necesară pentru a prelua orice document?** Dacă am avea o propagare a semnalului perfect regulată și $q = 1$, AP-urile dintr-un plan ar împărți spațiul între ele în **regiunile Voronoi**. Cel mai puternic AP ar fi centrul regiunii Voronoi, iar documentele punctele din regiunea Voronoi.

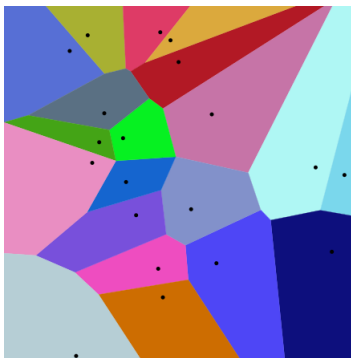


Figure 2: Regiunile Voronoi

În realitate, datorită propagării neregulate în interior, aceste regiuni nu au formă de Voronoi și nu au granițe definite, ci sunt parțial suprapuse. Forma și cantitatea suprapunerii depind de clădire, densitatea AP-urilor și metoda de colectare.

Complexitatea de căutare a acestei structuri de date este deci $O(q \star nV)$, unde nV este numărul de documente din interiorul unei regiuni Voronoi. Zonele

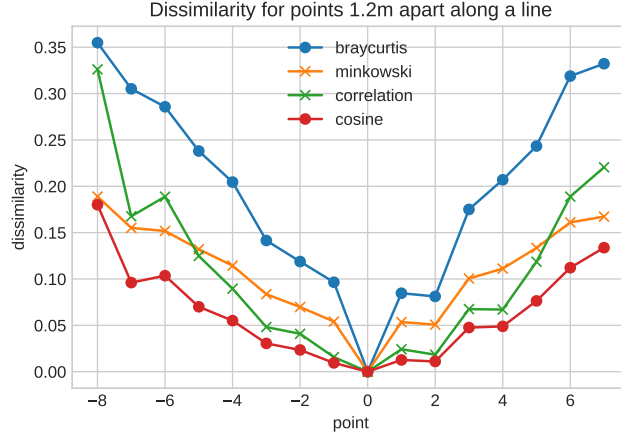


Figure 3: Dissimilarities of points on a line against point 0. RSSI compare function should behave monotonically for points close to the query point.

regiunilor Voronoi depind de densitatea AP-urilor (ca vârfuri Voronoi), dar din cauza valorilor RSSI variabile pentru cele mai puternice AP, dimensiunea regiunii va fi mai mare în practică. Acest proces de căutare este ușor paralelizabil, deoarece o semnătură poate fi căutată în paralel în fiecare dintre listele q .

Implementarea serverului este publică și se regăsește în repository-ul de github.¹ În acest repository am creat și un utilitar cu care administratorul serverului poate manipula baza de date cu documente și semnături: vizualizare, ștergere, inserare.

4 Rezultate măsurători

Semnalele WiFi au o propagare neregulată în interior datorită efectelor multipath și shadowing, construcției, mobilei, și persoanelor. Însă la o anumită locație propagarea către și de la access point, deși nu este predictibilă, are o stabilitate în timp și este specifică acelei locații. Același lucru este adevărat despre propagarea celulară 2G-5G, și despre propagarea sonoră a reflexiilor din pereți interiori. Totalitatea acestor semnale are o unicitate specifică locației unde se face recoltarea. Măsurarea semnalelor radio (BLE, WiFi, 2G-5G) se face oricum ca parte a protocoalelor uzuale pe telefoanele mobile, deci nu necesită un efort suplimentar de colectare. Noi propunem agregarea acestor semnale într-o ”semnătură” specifică locației fizice.

4.1 Măsura de disimilaritate

În general, poziționarea cu amprente radio folosește o funcție de distanță în spațiul semnal, de exemplu distanța euclidiană în articolul original RADAR[1], dar multe alte distanțe au fost propuse în literatură. Articolul [3] testează distanțele Minkowski, cosinus, Pearson correlation, și Shepard, arătând că distanțele euclidiană și corelarea Pearson produc cele mai bune rezultate. În alte studii, distanța Mahalanobis obține cea mai bună performanță, dar nu poate fi aplicată în cazul nostru, deoarece sistemul Android limitează scanarea la o citire per AP la 3-4 secunde, iar matricea de covariație între distribuțiile RSSI nu poate fi obținută fără așteptare prelungită.

Autorii din [7] explorează multe alte distanțe și disimilarități din literatură, arătând că Sørensen (coeficientul BrayCurtis) se comportă cel mai bine. Dintre acestea, în AirDocs am testat Minkowski ($p = 2$, Euclidean), cosine, corelația Pearson și Bray-Curtis.

Dacă considerăm puncte echidistante de-a lungul unei linii (Figura 3), se observă comportarea monotonă a disimilarității du distanța euclideană. Am ales să folosim mai departe Bray-Curtis deoarece obține o monotonicitate mai bună la testarea cu datele din setul 1. Deasemenea, am adoptat și alte îmbunătățiri propuse de [7]: normalizarea zero-unu (ecuația 1) a unei valori RSSI x_i în dBm:

$$X_i = \text{normalized}(x_i) = \alpha(1 - \frac{x_i}{\min})^e \quad (1)$$

Am ales o valoare α astfel încât plaja observată a lui x_i să fie în intervalul $[0,1]$. Scopul acestei normalizări este dublu: mapează valorile negative ale puterii (în dBm) la valori pozitive necesare pentru unele măsuri de similaritate, și reduce influența citirilor cu putere redusă. Aceasta din urmă duce la penalizarea mai puternică diferențelor între semnalele puternice, deoarece semnalele slabe au variații mari, așadar diferențele dintre ele sunt mai puțin semnificative. De exemplu, o diferență de la -90 dBm la -85 dBm este mai puțin importantă decât o diferență de la -40 dBm la -35 dBm.

Disimilaritatea Bray-Curtis (numită și distanță Sørensen), considerată ca fiind cea mai performantă în [7], se bazează pe AP-uri comune între cele două semnături:

$$BCurtis(X, Y) = \frac{\sum_{i=1}^c |X_i - Y_i|}{\sum_{i=1}^c (X_i + Y_i)} \quad (2)$$

Cele mai multe măsuri de di-similaritate consideră numai AP-urile comune, dar, așa cum arată Beder și Klepal [2], AP-urile ne-comune sunt un factor critic în a reduce similaritățile false. Pentru AP-urile lipsa între cele două semnături, considerăm ca ele sunt vizibile la -100 dBm (-99 dBm fiind minimul observat în data-seturile din acest studiu), astfel încât contribuie și ele la di-similaritate. În Figura 4, observăm că AP-urile slabe care sunt ne-comune nu afectează mult disimilaritatea, mai ales că e probabil ca telefoanele mobile să nu obțină semnalele slabe în mod fiabil. Prin contrast, AP-urile puternice care lipsesc sunt

¹<https://github.com/documenteinaer/ad-server>

un indicator pentru o locație care se află la distanță. O altă euristică aplicată în acest punct este și setarea dissimilarității la 1.0 dacă fracțiunea de AP-uri comune este sub un prag de 25%.

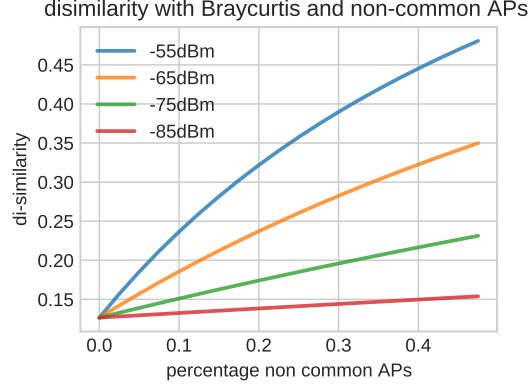


Figure 4: Disimilaritatea Braycurtis în cazul în care se include un procent de AP-uri ne-comune

4.1.1 Variabilitatea dispozitivelor folosite

Un alt aspect care trebuie considerat la evaluarea disimilarității este recoltarea semnăturilor cu dispozitive care au parametri diferiți pentru carduri și antenele mobile. Acestea sunt cauzate de folosirea unor antene cu pattern-uri de radiație diferite, lanțuri RF diferite. Pentru a neutraliza diferențele introduse de heterogeneitatea dispozitivelor, folosim o metodă de diferențiere derivată din lucrările [4] și [6], care au fost evaluată în [5] ca fiind eficientă la folosirea dispozitivelor diferite.

Pentru o amprentă WiFi, **only** pentru AP-urile comune $X = \{X_i\}$, invariantul semnăturii este obținut prin diferențierea puterii pentru AP-urile "consecutive":

$$X_{invariant} = \{(X_1 - X_2), (X_2 - X_3), \dots, (X_{n-1} - X_n), (X_n - X_1)\} \quad (3)$$

În AirDocs folosim o disimilaritate bazată pe Bray-Curtis, îmbunătățită cu toate caracteristicile menționate în această secțiune, și îi comparăm monotonicitatea cu privire la distanța euclidiană reală între puncte. În Figura 5, pentru datasetul 1, considerăm toate perechile de puncte de pe un etaj care sunt la 15 m sau mai aproape. Se observa că disimilaritatea crește liniar cu distanța, iar dincolo de pragul de 13 m, care este lungimea coridorului pătrat, începem să observăm valori ale disimilarității cu valoarea 1, deoarece mai mult de un sfert

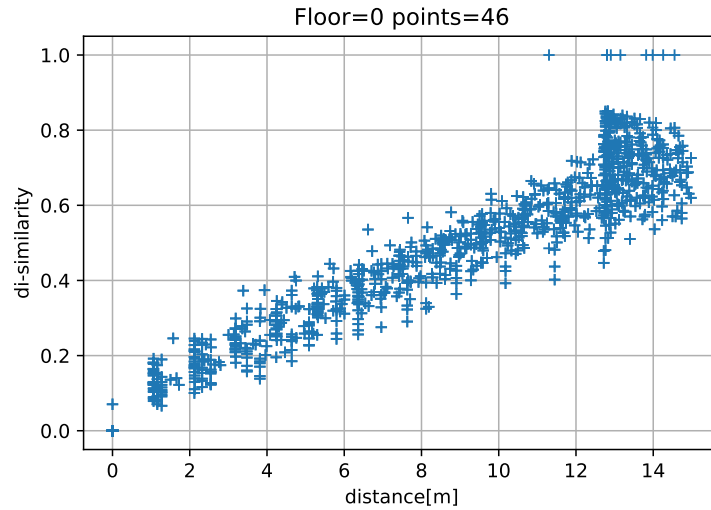


Figure 5: Braycurtis dissimilarity vs real distance.

dintre AP-uri devin ne-comune, si deasemenea o deviație mai mare datorită atenuării sporite a pereților.

5 Sumar

În această fază a proiectului, s-au îndeplinit următoarele obiective:

- implementarea aplicației Android: aplicația rulează pe telefoane care folosesc versiuni de Android cu versiunile 8, 10-12. O variantă a clientului pentru laptop-uri Linux este în dezvoltare
- implementarea serverului Python. Serverul poate rula pe orice mașină a instituției disponibilă în intranet sau în cloud.
- colectarea și analiza datelor. Folosind date măsurate în clădirile universității, dar și date publice disponibile am evaluat performanța sistemului pentru diverse condiții de deployment
- proiectarea metodelor de măsurare a similarităților între semnăturile colectate în clădiri: s-au adoptat norme de colectare și utilizare a semnăturilor pentru a obține o plasare și regăsire a documentelor cât mai performantă
- Publicație cu acknowledgment pentru proiect: L. Flueratoru, V. Shubina, D. Niculescu and E. S. Lohan, "On the High Fluctuations of Received Signal Strength Measurements with BLE Signals for Contact Tracing and Proximity Detection," in IEEE Sensors Journal, doi: 10.1109/JSEN.2021.3095710, Q2, IF 3.44

References

- [1] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000*, volume 2, pages 775–784. Ieee, 2000.
- [2] Christian Beder and Martin Klepal. *Fingerprinting Based Localisation Revisited a rigorous approach for comparing RSSI measurements coping with missed access points and differing antenna attenuations*. November 2012. Journal Abbreviation: 2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings Publication Title: 2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings.
- [3] Giuseppe Caso, Luca De Nardis, and Maria-Gabriella Di Benedetto. A mixed approach to similarity metric selection in affinity propagation-based wifi fingerprinting indoor positioning. *Sensors*, 15(11):27692–27720, 2015.
- [4] Fangfang Dong, Yiqiang Chen, Junfa Liu, Qiong Ning, and Songmei Piao. A calibration-free localization solution for handling signal strength variance. In Richard Fuller and Xenofon D. Koutsoukos, editors, *Mobile Entity Localization and Tracking in GPS-less Environments*, pages 79–90, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [5] Shih-Hau Fang, Chu-Hsuan Wang, Sheng-Min Chiou, and Pochiang Lin. Calibration-free approaches for robust wi-fi positioning against device diversity: A performance comparison. In *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2012.
- [6] A.K.M. Mahtab Hossain, Yunye Jin, Wee-Seng Soh, and Hien Nguyen Van. Ssd: A robust rf location fingerprint addressing mobile devices’ heterogeneity. *IEEE Transactions on Mobile Computing*, 12(1):65–77, 2013.
- [7] Joaquín Torres-Sospedra, Raúl Montoliu, Sergio Trilles, Óscar Belmonte, and Joaquín Huerta. Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems. *Expert Systems with Applications*, 42(23):9263–9278, 2015.