

Raportare proiect Airdocs, Etapa 1

Laura Ruse (Dragomir), Costin Carabaș, Dragoș Niculescu

November 2020

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

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, așa cum se exemplifică în figura 1.

Folosind același principiu, dorim explorarea altor senzor care pot oferi un context specific locației: sunet, câmp magnetic, busolă. Semnăturile unice obținute pot fi apoi folosite pentru gestionarea unei colecții de documente fără a fi necesară maparea acestor documente la poziții fizice corecte (longitudine, latitudine și altitudine). Se obține o asociere între document și poziția sa în clădire (dar nu o poziție carteziană corectă). Structura de date obținută se comportă ca o bază de date indexată pe baza locației, însă fără a necesita pozițiile reale care sunt costisitor de obținut și menținut.

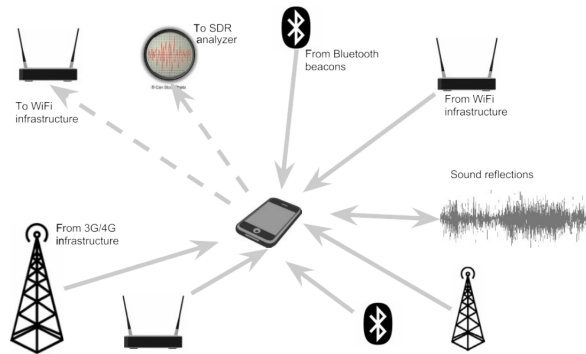


Figure 1: Crearea semnăturilor din surse accesibile telefoanelor mobile

Acest middleware va oferi trei primitive aplicațiilor (Figura 2):

- 1. $S = \text{CreateSgn}()$ 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
- 2. $\text{Put}(S, \text{document})$ 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).
- 3. $\text{Get}(S)$ - 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 acea locație. Serverul efectuează o căutare în spațiul semnăturilor, așadar locațiile geografice reale nu sunt necesare și nu sunt stocate în baza de date (dacă nu sunt disponibile).

Ce se obține cu acest sistem este o formă de realitate augmentată prin care utilizatorii au iluzia documentelor răspândite în spațiul fizic, vizibile doar la anumite locații. Plasarea unui document "în aer" permite un mod natural de a-l folosi ca o notă wireless pentru o etichetă la muzeu, o indicație de direcționare într-o clădire, aeroport sau mall, un anunț temporar pe o ușă, un meniu la restaurant, etc. Scopul proiectului AirDocs este de a experimenta cu aceste semnături bogate (creare, clasificare, mentenanță), cu căutarea în spațiul semnal, și de a implementa un demonstrator și aplicații Android care nu necesită un serviciu de localizare funcțional (în interior). Aceste aplicații de tipul wireless post-it, graffiti, navigare ghidată, de obicei ar necesita poziționare exactă, care este de cele mai multe ori dificilă în interior.

Aplicarea acestui middleware pe scară largă implică și implementarea unor politici specifice instituțiilor și clădirilor care doresc acest serviciu. De exemplu, diverși utilizatori pot avea permisiuni de postare în anumite locuri, pot avea restricții de dimensiune a documentelor, restricții de viață maximă a documentelor, sau alte aspecte de securitate care sunt specifice aplicației finale. Deasemenea aplicația finală poate folosi o interfață care să faciliteze producerea

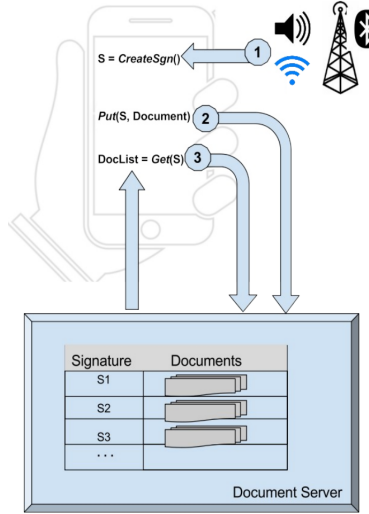


Figure 2: Primitivele Put si Get pe baza cărora sunt folosite semnăturile. Funcția Put plasează un document ”în aer” la o anumită locație în clădire identificată de semnătura S. Funcția Get regăsește documentele care au o semnătură asemănătoare cu S.

și consumul acestor documente într-o modalitate adecvată specifică aplicației. Aceste aspecte nu sunt scopul proiectului AirDocs, deși o parte din ele vor fi explorate în procesul de implementare al middleweare-ului propus.

2 State of the art

Asocierea dintre documente si locații a mai fost explorată - de exemplu în [32] documente web sunt etichetate cu locatia și referențiate pe baza locației. În comunitatea bazelor de date sunt eforturi de a formaliza căutarea spațială [38] a documentelor, ceea ce se poate implementa la granularitatea întregului Internet. Ideea de asocia POI(puncte de interes) hărtilor GPS pentru a asocia date unei locații fizice este utilizata în multiple aplicații pentru exterior acolo unde serviciul GPS este functional (sportive, jocuri, gamificarea vieții). Proiecte cum ar fi Digital Graffiti [12], de la Universitatea Linz/Austria, sau PlaceMail [22] doresc să asocieze date locațiilor exterioare, necesitând acces la GPS si la Internet. Un brevet CISCO recent [12] descrie o idee similară de a primi mesaje în diverse locatii fizice obținute folosind sistemul de localizare CISCO. Google Maps permite queryuri în jurul unei locații pentru a accesa POIuri gestionate de Google. Toate aceste metode necesită folosirea unei tehnologii de localizare, iar asocierea dintre date si poziții este stocată într-o bază de date de obicei sub controlul utilizatorului.

Pentru utilizarea în interior bazată pe antrenament, dificultatea constă munca

manuală necesară creării bazelor de date cu semnături radio corespunzătoare locațiilor.

O colecție de alte articole au fost investigate în problematicile: WiFi fingerprint (amprente radio), stocare documente în baze de date spațiale, și măsuri de similaritate cu date incomplete:

[7] [8] [11] [13] [24] [31] [12] [37] [28] [16] [39] [30] [5] [32] [27] [29] [39] [34] [15] [25] [1] [2] [22] [35] [18] [19] [6] [4] [14] [38] [3] [10] [7] [17] [33]

2.1 Date experimentale

Pentru a face experimente, este nevoie de o infrastructură, o clădire cu mai multe routere wireless. O posibilitate ar fi să folosim infrastructura din clădirea PRECIS din cadrul Universității POLITEHNICA din București. Pentru a eficientiza în primă fază colectarea acestor informații, am decis să folosim date [26] [21] deja colectate de alți cercetători. În aceste date există informații despre data calendaristică, dispozitivul de pe care s-a făcut colectarea, puterea conexiunii cu AP-urile, precum și coordonatele geografice. În [26] există informații colectate de la biblioteca Universității Jaume I (UJI) pe parcursul a 25 de luni, folosind un singur telefon cu Android și conține 103,584 semnături de Wi-Fi. În [21], s-au colectat date dintr-o clădire de 4 etaje din Universitatea Tampere din Finlanda. Colectarea s-a făcut pe parcursul a 8 luni și conține 4500 de semnături de Wi-Fi. Folosind cele două colecții de date, obținem diversitate prin numărul de dispozitive folosite, tipologia clădirii și putem vedea modificări ale datelor de-a lungul timpului (pe o perioadă maximă de 25 de luni).

Pentru a crea cu exactitate o semnătură asociată unei locații, trebuie să ținem cont de mai mulți factori: dispozitivul cu care se colectează (unele telefoane pot adăuga erori), sustenabilitatea în timp a unei semnături (două colectări la diferențe de 6 luni pot diferi). În [26], autorii menționează că pot exista 2-3 dBm diferență între două măsurători succesive. În Figura 3 putem vedea diferențele între 2 măsurători succesive: diferența între măsurările 1 și 2 este în prima coloană, între măsurările 2 și 3 în a doua coloană, șamd.

Sample Pair	1-2	2-3	3-4	4-5	5-6
Mean Difference (dBm)	3.19	2.06	1.82	1.74	1.95

Figure 3: Diferența de RSS între diferite puncte [26]

2.2 Deosebirea semnăturilor Wi-Fi

Pentru deosebirea semnăturilor trebuie să definim modalitatea de diferențiere. În [9] sunt prezentate mai multe metode de măsurare a distanței. Printre ele se numără Euclidean, Euclidean Standardizat, Mahalanobis, City block, Minkowski, Chebychev, Cosine, Correlation, Hamming, Jaccard, și Spearman.

Autorii demonstrează faptul că metodele City block, Chebychev, Euclidean, Mahalanobis, Minkowski, și Euclidean Standardizat au rezultate bune.

Pornind de la această [9] concluzie, am creat un spațiu cu N dimensiuni, unde N este numărul de AP-uri din colecția noastră [21] de date. Am măsurat distanțele folosind metodele City block, Euclidean, Minkowski, Cosine și Jaccard. Acestea se pot vedea în graficul din figura 4. Pe axa orizontală se află distanța față de un punct ales aleator și celelalte puncte din apropiere (20 de metri). Pe axa verticală se află o măsură a similarității conform fiecărei metode menționate anterior. Această valoare trebuie să fie proporțională cu distanța fizică între puncte. Putem observa că nu există o astfel de regulă.

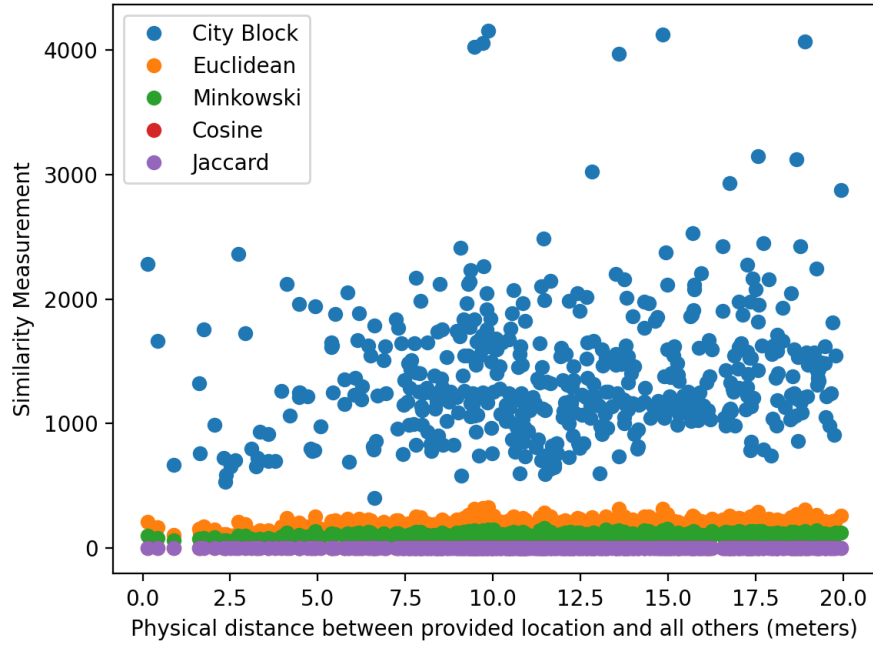


Figure 4: Măsura similarităților comparat cu distanțele fizice [21]

2.3 Date lipsă

Una din probleme este că există valori lipsă pentru fiecare locație. Din dimensiunea N , menționată anterior, există o submulțime lipsă a spațiilor. Compararea similarității pentru două semnături unde există valori lipsă nu este corectă.

În [23], autorii propun două măsuri de similaritate pentru soft-set-uri. În continuare vom verifica dacă formulele prezentate se aplică și pentru cazul nostru. Scopul final este de a compara două semnături distincte și de a le măsura

similaritatea.

În [36] autorii clasifică problema lipsei datelor în patru abordări. Printre cele mai importante sunt metoda eliminării spațiilor lipsă, metoda adăugării cu valori approximate, construirea unui model matematic și aproximarea acestor valori și în cele din urmă, folosirea tehnicilor de învățare automată. Vom încerca în continuare aceste metode.

În continuare am folosit metrica numărului de AP-uri detectate și numărul de AP-uri comune. Comparând o locație dată cu toate celelalte, cele apropiate vor avea mai multe AP-uri în comun decât cele mai depărtate. Putem observa în figura 5 că cele mai departe (aproape de 20 de metri distanță) au cele mai puține AP-uri în comun (aproape de 0), însă există diferențe mari în rest. De exemplu locații cu 120 de AP-uri comune se află atât la 5 metri distanță cât și la 20.

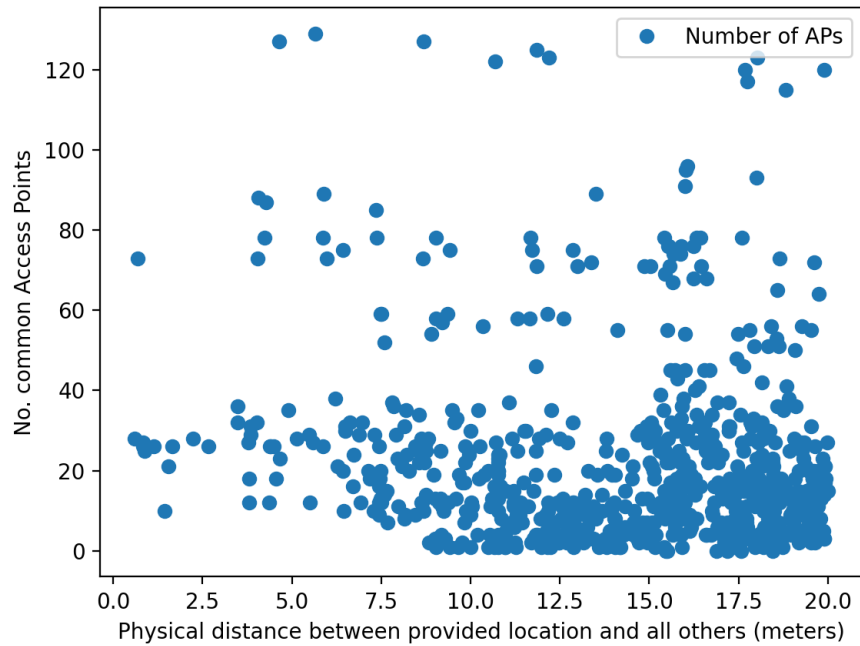


Figure 5: Numărul de Access Point-uri comune comparat cu distanța fizică

3 Colectarea datelor pe Android

Conceptul de bază de la care plecăm este faptul că puterea semnalului primit de la un dispozitiv este corelat cu distanța până la acel dispozitiv. Multe soluții

propușe în literatura de specialitate colectează puterea semnalului pentru a determina locația curentă. În cazul nostru, vom colecta puterea semnalului și alte informații și vom construi un fingerprint al locației unde vrem să plasăm sau de unde vrem să obținem un document.

O decizie importantă pe care a trebuit să o luăm a fost metoda de colectare a datelor. În primul rând, Android rulează pe telefoane mobile și tablete, dispozitive foarte portabile și ușor de folosit oriunde. Pe Android, în modul standard (fără să fie rootat), informațiile necesare sunt colectate de la serviciile de Android, la nivel aplicație. De aceea, nu avem control total asupra modului de scanare și frecvența acesteia. Am determinat experimental faptul că Android face o scanare completă a rețelelor WiFi în 3-4 secunde, după cum a fost determinat și în literatura de specialitate [20].

O alternativă ar fi fost folosirea unui laptop pentru colectarea datelor. Deși este un dispozitiv portabil, este mai greu de folosit oriunde și oricând. Totuși, dacă folosim Linux monitor mode, putem colecta date mai rapid și controla mai bine procesul de colectare. Dacă putem colecta mai multe date într-un interval mai scurt atunci putem obține o distribuție a RSSI-urilor per AP în loc de o singură valoare.

Din motive practice în viața reală, pentru ca utilizatorii să poată căuta documente oriunde și oricând, vom implementa clientul Airdocs pe dispozitive mobile, mai exact pe Android. Dar pentru experimente și determinarea celui mai bun algoritm de similaritate, vom colecta date și pe un laptop cu Linux.

O altă perspectivă a metodei de colectare se refera la cine colectează: 1) dispozitivul mobil, 2) AP-ul. Cel mai multe soluții din literatură [40], [3], [39] folosesc prima metoda, prin care dispozitivul mobil colectează puterea semnalului primit de la mai multe AP-uri. În acest caz nu avem nevoie de infrastructură specială, doar implementăm aplicația care colectează pe dispozitivul mobil. Dezavantajul este că sunt făcute operații pe dispozitivele mobile care pot consuma resursele acestora.

În al doilea caz, ca în soluția RADAR [4], AP-urile colectează puterea semnalului venit de la un anumit dispozitiv pentru a determina locația acestuia. În acest caz avem nevoie de o infrastructură specială, formată din AP-uri programabile. Avantajul este faptul că nu sunt făcute operații costisitoare pe dispozitivele utilizatorilor.

În literatura de specialitate putem observa faptul că datele colectate pot varia de la un dispozitiv la altul, chiar dacă sunt colectate în același loc și același moment.

În plus, datele pot varia în funcție de momentul zilei [3], în raport cu activitățile oamenilor din acel spațiu. Soluția noastră va trebui să ia în considerare aceste variații.

Arhitectura aplicației Airdocs este reprezentată în figura 6.

Pe Android a fost implementat un serviciu care colectează date despre rețelele WiFi, dispozitivele Bluetooth Low Energy (BLE), telefonie și coordonatele GPS.

Componenta WiFiScan folosește serviciul WifiManager din Android, ce oferă rezultatele unei scanări complete a rețelelor WiFi o dată la 3-4 secunde. De aceea am considerat aceasta fiind fereastra de colectare a unui fingerprint.

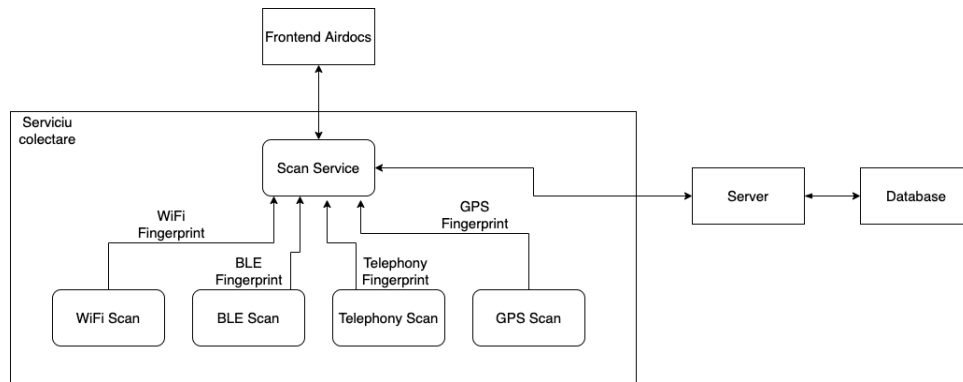


Figure 6: Arhitectura aplicației Airdocs

Pentru a încărca sau găsi un document, utilizatorul va trebui să stea pe loc pentru minim 3-4 secunde. Dacă este în mișcare, fingerprint-ul va putea include date invalide, deoarece în 3-4 secunde (cât este fereastra de colectare) va ajunge în altă locație, cu alt fingerprint.

O scanare completă a rețelelor WiFi include pentru fiecare rețea detectată, următoarele informații importante ce vor face parte din fingerprint: adresa hardware, SSID, frecvență, RSSI.

Componenta BLEScan folosește Bluetooth Adapter din Android. Descoperirea dispozitivelor Bluetooth Low Energy (BLE) se face în mod continuu. Un set de date include: numele dispozitivului (dacă este disponibil) și RSSI. Un fingerprint poate include mai multe seturi de date de la același dispozitiv BLE (primate în fereastra de colectare).

Componenta TelephonyScan folosește serviciul TelephonyManager pentru a colecta informațiile de la turnurile de telefonie mobilă. Un set de date include: CID (cell id) și RSSI. Un fingerprint poate include mai multe seturi de date cu același CID (primate în fereastra de colectare).

Componenta GPSScan folosește serviciul LocationManager pentru a obține coordonatele GPS. Acestea vor fi generate doar dacă dispozitivul mobil se află în apropierea unui geam, pentru a putea primi informații de la sateliții GPS. Un set de date include: latitudine, longitudine. Includem în fingerprint toate seturile de date primite în fereastra de colectare.

Fingerprint-ul include datele colectate în fereastra de 3-4 secunde. Fingerprint-ul este transformat în format JSON și este trimis la server în printr-un mesaj HTTP POST.

Momentan aplicația de Android este implementată pentru o versiune de Android de la 9 în sus deoarece de la API 29 s-a schimbat implementarea și API-ul TelephonyManager. Pe viitor se poate face o implementare alternativă a TelephonyScan care să funcționeze și pe versiuni anterioare.

Serverul va fi implementat în Python și va include următoarele componente:
1) un server HTTP minimal ce primește mesaje HTML POST ce includ un

JSON; 2) Componenta de parsare a JSON-ului; 3) componenta de procesare a datelor primite; 4) componenta de calcul al similarității; 5) componenta de comunicare cu baza de date.

Arhitectura serverului Airdocs este reprezentată în figura 7.

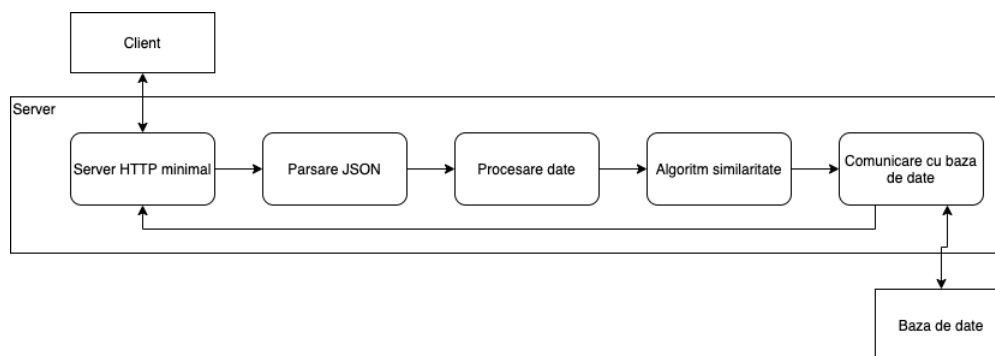


Figure 7: Arhitectura serverului Airdocs

Implementarea serverului va trebui să fie scalabilă la un număr mare de cereri de găsim a documentelor, la un număr mare de utilizatori, la un număr mare de AP-uri și la un număr mare de documente.

4 Sumar

Proiectul Airdocs a demarat cu investigarea disponibilității dispozitivelor Android pentru colectarea semnăturilor radio, a măsurilor de similaritate, folosirea unor date publice pentru a testa măsurile de similaritate, și cu implementarea unui prototip de colectare pe Android.

References

- [1] Microsoft indoor localization competition, 2018. <https://www.microsoft.com/en-us/research/event/microsoft-indoor-localization-competition-ipsn-2018/>.
- [2] Microsoft indoor localization competition, 2018. <https://www.kaarta.com/kaarta-wins-microsoft-indoor-localization-competition-for-second-consecutive-year/>.
- [3] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272, 2009.

- [4] 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.
- [5] Christian Böhm, Stefan Berchtold, and Daniel A Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.
- [6] Cristina Bonanni, Paola Carlesimo, Giovanni BP Catasta, Giuseppe Longobardi, and Alessandra Rotunno. Proximity based reminders, January 26 2016. US Patent 9,247,387.
- [7] Giuseppe Caso, Luca De Nardis, Filip Lemic, Vlado Handziski, Adam Wolisz, and Maria-Gabriella Di Benedetto. Vifi: Virtual fingerprinting wifi-based indoor positioning via multi-wall multi-floor propagation model. *IEEE Transactions on Mobile Computing*, 19(6):1478–1491, 2019.
- [8] Dimitris Chatzopoulos, Carlos Bermejo, Zhanpeng Huang, and Pan Hui. Mobile augmented reality survey: From where we are to where we go. *Ieee Access*, 5:6917–6950, 2017.
- [9] Kittipong Chomboon, Pasapitch Chujai, Pongsakorn Teerarassammee, Kittisak Kerdprasop, and Nittaya Kerdprasop. An empirical study of distance metrics for k-nearest neighbor algorithm. pages 280–285, 01 2015.
- [10] Andrei Croitoru, Dragos Niculescu, and Costin Raiciu. Towards wifi mobility without fast handover. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pages 219–234, 2015.
- [11] Ramsey Faragher and Robert Harle. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.
- [12] Peter Michael Gits and Dale Seavey. Delivering wireless information associating to a facility, March 1 2016. US Patent 9,277,374.
- [13] Kang Eun Jeon, James She, Perm Soonsawad, and Pai Chet Ng. Ble beacons for internet of things applications: Survey, challenges, and opportunities. *IEEE Internet of Things Journal*, 5(2):811–828, 2018.
- [14] Donnie H Kim, Younghun Kim, Deborah Estrin, and Mani B Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 43–56, 2010.
- [15] Kyu-Han Kim, Kang G Shin, and Dragos Niculescu. Mobile autonomous router system for dynamic (re) formation of wireless relay networks. *IEEE Transactions on Mobile Computing*, 12(9):1828–1841, 2012.

- [16] Mikkel Baun Kjærgaard. A taxonomy for radio location fingerprinting. In *International symposium on location-and context-awareness*, pages 139–156. Springer, 2007.
- [17] Hyuk Lim, Lu-Chuan Kung, Jennifer C Hou, and Haiyun Luo. Zero-configuration indoor localization over ieee 802.11 wireless infrastructure. *Wireless Networks*, 16(2):405–420, 2010.
- [18] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. Push the limit of wifi based localization for smartphones. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 305–316, 2012.
- [19] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.
- [20] Hung-Huan Liu and Chun Liu. Implementation of wi-fi signal sampling on an android smartphone for indoor positioning systems. *Sensors*, 18(2):3, Dec 2017.
- [21] Elena Simona Lohan, Joaquin Torres-Sospedra, Philipp Richter, Helena Leppäkoski, Joaquin Huerta, and Andrei Cramariuc. Crowdsourced wifi database and benchmark software for indoor positioning, Sep 2017.
- [22] Pamela J Ludford, Dan Frankowski, Ken Reily, Kurt Wilms, and Loren Terveen. Because i carry my cell phone anyway: functional location-based reminder applications. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 889–898, 2006.
- [23] Pinaki Majumdar and Syamal Samanta. Similarity measure of soft sets. *New Mathematics and Natural Computation (NMNC)*, 04:1–12, 03 2008.
- [24] Prodromos Makris, Dimitrios N Skoutas, and Charalabos Skianis. A survey on context-aware mobile and wireless networking: On networking and computing environments’ integration. *IEEE communications surveys & tutorials*, 15(1):362–386, 2012.
- [25] Thomas L Marzetta. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE transactions on wireless communications*, 9(11):3590–3600, 2010.
- [26] Germán Martín Mendoza-Silva, Philipp Richter, Joaquín Torres-Sospedra, Elena Simona Lohan, and Joaquín Huerta. Long-term wi-fi fingerprinting dataset and supporting material, Aug 2018.
- [27] Dragoş Niculescu. Interference map for 802.11 networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 339–350, 2007.

- [28] Dragos Niculescu. Finding mimo. In *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, pages 267–272. IEEE, 2010.
- [29] Cătălin Nicutar, Dragoș Niculescu, and Costin Raiciu. Using cooperation for low power low latency cellular connectivity. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 337–348, 2014.
- [30] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal De Lara. Accurate gsm indoor localization. In *International conference on ubiquitous computing*, pages 141–158. Springer, 2005.
- [31] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414–454, 2013.
- [32] Ingmar Poesse, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. Ip geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review*, 41(2):53–56, 2011.
- [33] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304, 2012.
- [34] Rathin Chandra Shit, Suraj Sharma, Deepak Puthal, and Albert Y Zomaya. Location of things (lot): A review and taxonomy of sensors localization in iot infrastructure. *IEEE Communications Surveys & Tutorials*, 20(3):2028–2061, 2018.
- [35] Timothy Sohn, Kevin A Li, Gunny Lee, Ian Smith, James Scott, and William G Griswold. Place-its: A study of location-based reminders on mobile phones. In *International Conference on Ubiquitous Computing*, pages 232–250. Springer, 2005.
- [36] Cao Truong Tran, Mengjie Zhang, Peter Andreae, and Bing Xue. Improving performance for classification with incomplete data using wrapper-based feature selection. *Evolutionary Intelligence*, 9, 09 2016.
- [37] Yu-Chih Tung and Kang G Shin. Echotag: Accurate infrastructure-free indoor location tagging with smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 525–536, 2015.
- [38] Dingming Wu, Gao Cong, and Christian S Jensen. A framework for efficient spatial web object retrieval. *The VLDB Journal*, 21(6):797–822, 2012.
- [39] Zheng Yang, Chenshu Wu, and Yunhao Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 269–280, 2012.

- [40] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, page 205–218, New York, NY, USA, 2005. Association for Computing Machinery.