SmartPark : Se garer dans les Smart Cities ? Un bon créneau pour les smartphones.

Jean-Gabriel Krieg[1], Gentian Jakllari[1], Hadrien Toma, André-Luc Beylot[1]

[1] *Université de Toulouse - IRIT*

De plus en plus connectées, les villes recherchent un système de détection de places de parking sur smartphone. Pourtant, les coûts de déploiement et de maintenance de l'infrastructure nécessaires aux solutions existantes demeurent beaucoup trop élevés. Nous présentons SmartPark, un système de stationnement en ville sans infrastructure dédiée. SmartPark utilise un système de classification de type Random Forest à partir des capteurs d'un smartphone pour déterminer la libération d'une place de parking. Nous proposons par ailleurs une nouvelle métrique de comparaison de signaux RF environnants, afin de résoudre le problème du changement de véhicule.
Nous avons mené des expérimentations à l'aide de 12 volontaires, utilisant 7 smartphones différents et dans 3 villes différentes. Elles ont montré que SmartPark peut identifier 9 modes de transport différents avec une précision de 98,72%, ce qui permet la détection de la libération d'une place de parking dans 29 cas sur 29.

Keywords: parking, Smart City, classification, transportation, sensors, smartphone, Wi-Fi

1 Motivations et état de l'art

La recherche d'une place de parking libre en ville est un réel problème qui engendre 30% du trafic urbain [Sho06], fait perdre un temps précieux aux utilisateurs et augmente l'empreinte carbone des villes. De plus, une application pour le stationnement pourrait permettre aux villes de proposer un paiement en ligne, et ainsi éviter les impayés. La solution de Google, Open Spot, a connu un relatif échec car elle nécessitait une intervention importante de l'utilisateur. Par la suite, la solution du signal GPS a été proposée [SWYX11] et a montré une précision de plus de 90%. Cette solution, a priori efficace, a montré ses limites : la consommation de batterie qui en résulte ne permet pas d'envisager une utilisation à grande échelle. Plusieurs villes, comme San Francisco, ont alors installé une infrastructure spécifique, solution coûteuse en termes de déploiement et de maintenance. [NEM] a été le premier système capable de relâcher à la fois les contraintes de consommation de batterie et d'absence d'infrastructure, mais demande toujours une action importante des utilisateurs.

Nous proposons donc SmartPark, un système de détection de places de parking, sans besoin d'infrastructure, sans besoin d'action importante de l'utilisateur et avec un impact mineur sur la batterie de son smartphone. SmartPark fonctionne en 4 phases.
1) L'utilisateur recherche une place de parking. SmartPark détecte automatiquement le type de véhicule de l'utilisateur - piéton, vélo, bus, voiture, métro, moto, train, tramway, avion -, et envoie une requête de places libres pour ce type de véhicule au serveur via le réseau disponible.
2) L'utilisateur choisit une des places proposées et se gare : SmartPark scanne directement les réseaux Wi-Fi et cellulaires à portée afin de créer un profil de localisation.
3) Le véhicule est garé. L'utilisateur peut vaquer à ses occupations, marcher, prendre le bus, le tramway, le métro ou n'importe quel mode de transport. Pendant ce temps, toutes les 10 s, SmartPark détermine le mode de transport de l'utilisateur, et déclenche une alerte si l'utilisateur se retrouve dans le même type de véhicule que détecté en 1).
4) L'utilisateur libère sa place de parking. SmartPark détecte cet événement grâce à 3). Afin de vérifier qu'il a repris sa voiture garée, SmartPark réalise un nouveau scan réseau. S'il est semblable au premier, SmartPark envoie une notification de libération de place de parking au serveur. Sinon, SmartPark revient au point 2).

Ce papier est une version courte de [KJTB16].

2 Détection du mode de transport

Dans cette section, nous présentons le module de détection de mode de transport. Il nécessite seulement 2 s de lecture des capteurs d'un smartphone, puis trouve le mode de transport de l'utilisateur du mobile parmi : **piéton, vélo, bus, voiture, métro, moto, train, tramway, avion**.

2.1 Utilisation de plusieurs capteurs

La Figure 1 représente la répartition du niveau sonore moyenné sur 2 s dans les différents modes de transport considérés en utilisant un LG Google Nexus 5. On peut constater que le niveau sonore moyen à moto est plus élevé qu'en bus ou à pied. En revanche, les valeurs sont proches et plusieurs modes de transport peuvent amener au même niveau sonore, ce qui rend difficile la détection du mode de transport sur ce seul paramètre. La Figure 2 montre un phénomène similaire sur l'accélération frontale par rapport à l'utilisateur. On peut notamment observer une forte diversité de valeurs pour la voiture, alors que la boxplot est plus resserrée pour le piéton, qui montre une accélération plutôt faible, ce qui correspond aux limites biologiques.

Prenant en compte ce constat, SmartPark combine les signaux de plusieurs capteurs, pour essayer d'obtenir une meilleure classification. Par exemple, la Figure 3 montre qu'en combinant les deux capteurs présentés précédemment, on peut dégager des zones pour trois modes de transport : voiture, avion et train.

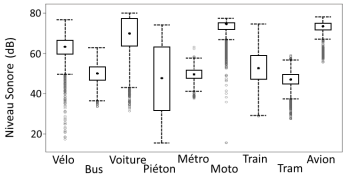


Figure 1: "Boîte à moustaches" du niveau sonore dans les différents modes de transport.

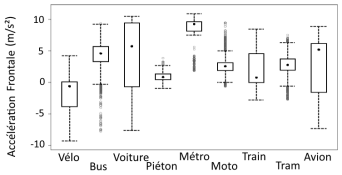


Figure 2: "Boîte à moustaches" de l'accélération frontale sur les différents modes de transport.

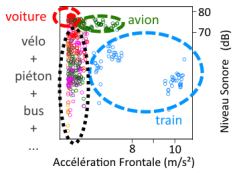


Figure 3: Projection des valeurs des capteurs de son et d'accélération pour différents modes de transport.

Pourtant, deux capteurs ne suffisent pas pour pouvoir distinguer tous les modes de transport envisagés. C'est la raison pour laquelle SmartPark combine les signaux de tous les capteurs communs d'un smartphone : l'accéléromètre, le gyroscope, le capteur d'orientation, le magnétomètre (pour tous ceux-là, dans les trois directions), le capteur de lumière, le baromètre et un capteur de niveau sonore (codé à partir des relevés du microphone). Tous ces capteurs se trouvent dans la grande majorité des smartphones récents.

2.2 Classification

Comme le montre le Tableau 1, SmartPark retient une totalité de 64 statistiques dans chaque domaine temporel, fréquentiel, par ondelettes de l'ensemble des capteurs. Cela conduit à 2 880 variables explicatives étudiées.

Domaines	Temporel, frequentiel, ondelettes
Statistiques générales	Moyennes, écart type, variance, médiane, valeur efficace, quartiles
5 pics et 5 vallées	Valeur, magnitude, largeur à mi-hauteur, 2 inter-distances
Autres	Kurtosis, asymétrie, entropie, surface

Tableau 1: Statistiques extraites de chaque capteur pour créer les variables du modèle.

Nous avons choisi d'utiliser une méthode de classification par arbre, plus rapide et plus simple à parcourir sur des smartphones en temps réel, et surtout très adaptée à des variables explicatives hétérogènes comme dans notre cas. À chaque nœud de l'arbre, l'algorithme cherche la variable explicative et un seuil qui permet de diviser les observations en deux branches les plus semblables - c'est-à-dire les moins désordonnés. Dans notre contexte, l'ordre correspond au fait d'avoir le plus faible nombre de modes de transport différents dans chaque branche. L'entropie est donc calculée à chaque nœud i : $D_i = -\sum_{m \in \forall modes} \frac{n_{m,i}}{N_i} log(\frac{n_{m,i}}{N_i})$

où N_i est le nombre d'observations dans le nœud i et $n_{m,i}$ le nombre d'observations correspondant au mode de transport m dans le nœud i. Finalement, le nœud final attribue le mode de transport le plus probable, c'est-à-dire celui dont le nombre d'observations est le plus grand.

Afin d'utiliser le plus efficacement possible l'ensemble des signaux des variables explicatives, SmartPark utilise un algorithme de type Random Forest (FR) [Bre01] qui crée plusieurs arbres. Pour chaque arbre, à chaque nœud, la meilleure variable explicative est sélectionnée parmi une sous-partie aléatoirement tirée de l'ensemble des variables. En répétant l'opération sur un grand nombre d'arbres - 500 pour SmartPark -, on obtient un grand nombre d'arbres de décisions décorrélés. Ces arbres sont déterminés hors-ligne sur une série d'observations où l'on connaît le mode de transport des signaux considérés. Ensuite, lorsque l'application fonctionne, elle n'aura qu'à parcourir les arbres jusqu'au nœud final, ce qui est une opération plutôt simple pour un smartphone récent.

3 S'agit-il vraiment du même véhicule?

Quand un utilisateur gare son véhicule (voiture, moto ou vélo), il entre le numéro de la place de parking dans son application SmartPark. Automatiquement, SmartPark crée un profil de localisation en réalisant un scan Wi-Fi et un scan du réseau cellulaire. On obtient le numéro et le niveau de signal reçu des points d'accès Wi-Fi et des cellules à portée. Dès lors, lorsque SmartPark détecte que l'utilisateur est revenu dans le même mode de transport que le véhicule qu'il a garé, il réalise un nouveau scan, et vérifie que le profil de localisation est semblable à celui stocké lorsqu'il s'est garé.
Nous proposons alors un coefficient de similarité C :

$$C = \frac{1}{\sum_{i=1}^{a} Ws_1(i)} \sum_{i=1}^{a} Ws_1(i)(1 - |\frac{Ws_1(i) - Ws_2(i)}{Ws_1(i)}|) + \frac{1}{\sum_{i=1}^{b} Wc_1(i)} \sum_{i=1}^{b} Wc_1(i)(1 - |\frac{Wc_1(i) - Wc_2(i)}{Wc_1(i)}|)$$

où $Ws_j(i)$ and $Wc_j(i)$ représentent les niveaux de signal du scan j pour respectivement le point d'accès et la cellule i. a est le nombre de points d'accès captés lors du premier scan (au moment du stationnement) et b le nombre de cellules.
Nous avons créé ce coefficient pour qu'il tienne en compte trois principes : la proportion de points d'accès trouvés ; la valeur des niveaux de signal - plutôt que d'utiliser un modèle binaire, plus la différence entre les deux mesures d'un même point d'accès est faible, plus le coefficient est grand - ; la proximité des points d'accès - plus le point d'accès est proche, plus la puissance reçue est grande, plus son poids dans le coefficient est important.

4 Performances

Pour réaliser une étude de performances de SmartPark, nous avons conduits des expérimentations sur 12 volontaires, dans 3 villes différentes et avec 7 smartphones différents (Sony Xperia E3, LG Google Nexus 5, LG Google Nexus 6, LG G2, Huawei Honor 4X, Samsung Galaxy S5, Samsung Galaxy S3 mini). Nos résultats sont comparés à ParkSense [NEM], Reddy [RMB+10], Stenneth [SWYX11], Zheng [ZLC+], Wang [WCM] et Peaks [HNT].
Nous avons demandé aux 12 volontaires d'entrer régulièrement dans notre application leur mode de transport dans leur vie quotidienne. SmartPark enregistre les données capteurs à la plus grande fréquence disponible, généralement 200Hz, calcule les variables explicatives, puis déduit des arbres de classification le mode de transport du conducteur. Nous avons obtenu finalement environ 28 heures d'expérimentations et nous avons comparé les 10 282 décisions de SmartPark au mode de transport entré par nos volontaires. Le Tableau 2 affiche les précisions obtenues. SmartPark détecte chaque mode de transport avec une précision d'au moins 95%. En moyenne, la précision obtenue est de 98,72%.

Transport	Vélo	Bus	Voiture	Métro	Moto	Piéton	Train	Tram	Avion	Motorisé
SmartPark	99%	98%	99%	96%	98%	100%	100%	99%	99%	99%
Peaks	-	78%	-	65%	-	95%	68%	84%	-	74%
Wang	81%	58%	72%	53%	-	81%	-	-	-	54%
Reddy	87%	-	-	-	-	95%	-	-	-	95%
Zheng	66%	67%	86%	-	-	89%	-	-	-	76%
Stenneth	89%	88%	88%	-	-	98%	98%	-	-	91%

Tableau 2: SmartPark permet une meilleure précision dans la détection du mode de transport que les systèmes existants.

Nous avons demandé ensuite à un volontaire d'utiliser SmartPark durant plusieurs jours à chaque fois qu'il se garait. Nous lui avons ensuite demandé de réaliser des "fausses libérations de place" : il gare son véhicule, puis monte dans le véhicule d'un ami un peu plus loin. En tout, le volontaire a réalisé 29 libérations de place et 27 fausses. Nous avons alors comparé les résultats de SmartPark avec les deux coefficients proposés par [NEM], le coefficient Tanimoto et l'indice de Jaccard, utilisés communément pour la comparaison de scans Wi-Fi. La Figure 4 montre les résultats pour différentes valeurs de seuils. L'indice de Jaccard est trop sévère dans notre cas et ne reconnaît pas la similarité des localisations lors de la libération de la place. Tanimoto et Parksense fonctionnent mieux. Cependant, quelque soit le seuil, ils conduisent à un grand nombre de faux positifs. SmartPark identifie 97% des événements de libération, sans aucun faux-positif pour un seuil de 0,5.

Nous avons enfin demandé à 7 volontaires dans 3 villes différentes d'utiliser SmartPark pendant 2 mois. Il leur est demandé de noter leurs événements de "parking" et de "libération" dans l'application, en ajoutant leur moyen de transport. À la fin de l'expérience, les données ont montré que les volontaires se sont garés 27 fois en voiture et 2 fois en vélo. SmartPark a détecté correctement 100% de ces événements, sans faux

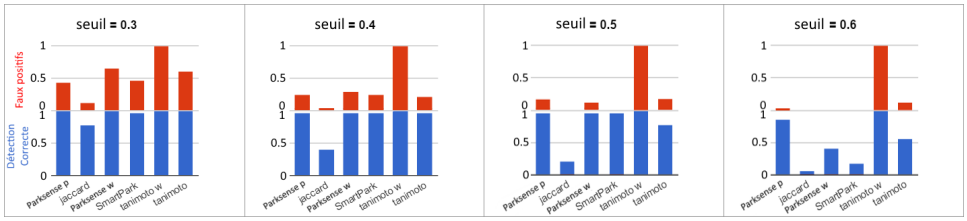


Figure 4: Taux de détection correcte (en bas, en bleu) et taux de fausse alarme (en haut, en rouge).

positif. Nous avons pu confirmer par ailleurs que l'application ne consomme pas énormément d'énergie : 1% en moyenne par heure sur un LG Nexus 5.

5 Conclusions

Nous proposons SmartPark, un système qui repose exclusivement sur les capteurs d'un smartphone pour indiquer aux conducteurs des places de parking libres, et un paiement automatique via l'application suivant le temps de stationnement. Pour cela, SmartPark résoud le problème de la détection de la libération d'une place de parking par un module de détection de mode de transport et un coefficient de vérification de localisation. La solution est fondée sur une analyse statistique approfondie des signaux des capteurs de smartphone et une classification de type Random Forest à partir de 2 880 variables explicatives. 9 modes de transport sont considérés et permettent la détection en temps réel du mode de transport par le smartphone. Nous avons évalué SmartPark sur 7 modèles de smartphones, dans 3 villes et à l'aide de 12 volontaires, et nous avons montré 95% de détection correcte du mode de transport, ce qui mène à 100% de détection de libération de places sur 29 événements de parking.

References

[Bre01] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[HNT] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *ACM SenSys '13*.

[KJTB16] JG. Krieg, G. Jakllari, H. Toma, and AL. Beylot. Unlocking the Smartphone's Senses for Smart City Parking. In *IEEE International Conference on Communications*, 2016.

[NEM] S. Nawaz, C. Efstratiou, and C. Mascolo. ParkSense: A Smartphone Based Sensing System for On-street Parking. In *ACM MobiCom '13*, pages 75–86.

[RMB+10] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27, 2010.

[Sho06] D.C. Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, 2006.

[SWYX11] L. Stenneth, O. Wolfson, P.S. Yu, and B. Xu. Transportation mode detection using mobile phones and gis information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, volume 37, pages 54–63. ACM, 2011.

[WCM] S. Wang, C. Chen, and J. Ma. Accelerometer based transportation mode recognition on mobile phones. In *Proceedings of the 2010 Asia-Pacific Conference on Wearable Computing Systems*, APWCS '10.

[ZLC+] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *ACM UbiComp '08*.