Towards Reliable Spatial Information in LBSNs

Ke Zhang
University of Pittsburgh
kez11@pitt.edu

Wei Jeng
University of Pittsburgh
wej9@pitt.edu

Francis Fofie
University of Pittsburgh
fof1@pitt.edu

Konstantinos Pelechrinis
University of Pittsburgh
kpele@pitt.edu

Prashant Krishnamurthy
University of Pittsburgh
prashant@sis.pitt.edu

ABSTRACT

The proliferation of Location-based Social Networks (LBSNs) has been rapid during the last year due to the number of novel services they can support. The main interaction between users in an LBSN is location sharing, which builds the spatial component of the system. The majority of the LBSNs make use of the notion of check-in, to enable users to volunteeringly share their whereabouts with their peers and the system. The flow of this spatial information is unidirectional and originates from the users' side. Given that currently there is no infrastructure in place for detecting fake check-ins, the quality of the spatial information plane of an LBSN is solely based on the honesty of the users. In this paper, we seek to raise the awareness of the community for this problem, by identifying and discussing the effects of the presence of fake location information. We further present a preliminary design of a fake check-in detection scheme, based on location-proofs. Our initial simulation results show that if we do not consider the infrastructural constraints, location-proofs can form a viable technical solution.

Author Keywords

Location-based social networks, Location proofs, Security.

ACM Classification Keywords

H.0 Information Systems: General.; K.6.5 Management of Computing and Information Systems: Security and Protection.

General Terms

Design, Reliability, Security

INTRODUCTION

Location-based Social Networks (LBSN) have attracted a lot of attention during the last years. While they exist since the early 2000s (e.g., Dodgeball was founded in 2003), it is only recently that LBSNs have taken off, mainly due to the advancements in mobile handheld devices. The latter allow for a fairly accurate positioning, thus, forming an ideal platform for the realization of advanced location sharing applications.

An LBSN has two distinct components: a social network and a location log for each user. The social part of the system resembles any other existing online social network, where friendships are declared and people can interact with their friends. What differentiates LBSNs for any other digital social network is the type of interaction that are feasible among the users. The main feature of this interaction is location sharing. Users volunteeringly share their location with their friends (or even with everyone in the system depending on the privacy settings). This location information can be either in the form of a trajectory continually tracked by the provider (e.g., systems such as Loopt) or in the form of volunteering sharings of the actual place/venue the user is in through a *check-in* (e.g., systems such as Foursquare). Some systems might also offer both alternatives (e.g., Google Latitude). Clearly, the second approach, where locations are tagged with semantic information (e.g., "I am in the Starbucks") as compared with a geographic trajectory (e.g., specific latitude/longitude), offers richer data that can enable novel services. Hence, this is the model that most popular LBSN utilize. A nice overview of location-based social networks and systems in general can be found in [18] for the interested reader.

In both of the aforementioned models, the flow of the spatial information is unidirectional. In particular, the user provides his location to the system, and as a consequence to the rest of the network. In the above process there is no proof of correctness for any information provided. However, as He *et al.* have shown [8], it is very easy to interfere with the positioning system of a mobile device and alter it in order to report fake coordinates. Moreover, in checkin-based LBSNs the users do not even have to alter the GPS' API to forge their whereabouts; they can simply bypass the automatic localization module[1] and check-in at a different venue than the one they actually are. While some LBSNs offer basic schemes to identify fake check-ins (e.g., the cheater code of foursquare [1]), their scope is limited (e.g., they do not perform well when the dishonest user is located fairly close to the venue he claims to be in).

UbiComp '12, Sep 5-Sep 8, 2012, Pittsburgh, USA.

[1]Since the accuracy and/or the availability of the GPS can be low, especially in urban areas, LBSNs allow users to manually enter the required information if needed.

While in this paper we are not interested into identifying the reasons behind location cheating, the incentives for adopting similar behaviors vary and can be present in a high degree. LBSNs have a strong gaming component and many users are interested in these mobile games [11]. Hence, they might be inclined to cheat simply to gain more virtual rewards (e.g., more points, *badges/pins*, etc.). Moreover, while LBSNs were designed mainly with the objectives of connecting people in space, helping them to meet new people in their vicinity, keep track of their own friends and explore new areas, business-related features and interactions related to monetary gains have recently taken-off. For instance, the owner of a venue can offer deals to users that check-in his venue [4]. The majority of these offers (more than 90%) require multiple check-ins [8]. Hence, a user, say Jack, can create a number of fake check-ins for unlocking this offer easier, leading to a monetary loss for the venue owner. In another context, Jack might also be tempted to share a fake location with the system in order to provide some sort of "alibi" or mislead other people with regards to his location.

The contribution of our work is two-fold and can be summarized in the following: **(i)** Raise the awareness of the community for the importance of identifying fake location sharings in LBSNs. We emphasize on the importance of solving this problem by discussing the effects of counterfeit spatial information. **(ii)** Design of a preliminary system, based on the primitives of location proofs, for the detection of fake check-ins. To the best of our knowledge this is the first scheme to tackle this problem.

The rest of the paper is organized as follows. Section discusses studies related with our work. Section analyzes possible effects of the presence of forged check-ins, while Section presents the design and evaluation of our preliminary detection scheme. Finally, Section concludes our work.

RELATED STUDIES

With the increased importance of spatial information for various applications, location-proofs have gained attention in the research community during the last years. Denning and MacDoran [6] describe a location-based authentication system where the position at any time is uniquely identified by a location signature. The signature is created by a location signature sensor (LSS) and it is time varying, hence, making it difficult to be forged. However, this system relies on a dedicated hardware and requires auxiliary equipments to strengthen the weak GPS signal in indoor environment. Saroiu and Wolman [14] design a scheme where location proofs are handed out by WiFi access points (APs). Each mobile device signs the APs' beacons and send them back to APs. The latter upon reception of the signed beacon creates a location signature for the mobile user. Zhang *et al.* [17] also utilize WiFi infrastructure and design a power modulated challenge-response location verification system. This mechanism utilizes RF signal strength from multiple APs to verify whether the claimed location is within the overlapping range of neighbouring APs. Furthermore, Kjærgaard and Wirz [9] present a clustering approach to detect indoor flocks of mobile users (i.e., spatio-temporal clusters).

In the context of LBSN, as aforementioned He *et al.* [8] have identified the problem of fake check-ins, without providing any solution to it. Foursquare has developed the *cheater code* [1] in an effort to minimize fake spatial information. The cheater code imposes additional rules on users' check-in frequency and speed. However, this mitigates potential fake check-ins to some extent only, since cheaters can easily bypass this detection [8]. Furthermore, location learning schemes can potentially used to enhance the detection of fake check-ins. For instance, Lian and Xie [10] propose a scheme to identify the location of a specific check-in based on primitives of location search. The location identified from the system can then be compared with the one claimed from the user. However, this scheme will be able to identify a number only of non-sophisticated fake check-ins (e.g., users that check-in to a remote locale without altering their GPS coordinates).

Our preliminary system design, is based on the primitives of location-proofs and can be complementary to efforts such as the cheater code. The key point is that a cheating user, say Jack, while being able to fake his GPS coordinates, he cannot do the same with the wireless channel's propagation characteristics. In brief, we utilize the notion of location signature using WiFi infrastructure enhancing it with the notion of flocks for identifying users that are not at the location (at the time) they claim to be at.

THE EFFECTS OF FAKE CHECK-INS

Traditionally, social information systems and information quality have followed disjoint paths. However, the presence of low quality of information (QoI) results in a decreased value for the specific platform. For instance, a very representative type of social information systems, vulnerable to low QoI is the Q&A social networks [16] [13]. In these systems, people post questions that can be answered from their peers, and are focused on providing an efficient platform for enabling the crowdsourcing nature of the underlying system. However, no care is taken for the actual quality of the answers provided. In the case of Q&A networks, feedback from users can be roughly used as a metric of the quality of the answer provided.

While similar issues exist related to the location sharing in LBSNs, there are no systems to date that are able to filter fake check-ins to a great extent. However, the existence of forged spatial information in the network can have significant effects in a wide spectrum of the underlying functionalities. In this section we will discuss two representative examples; one related to the effects on participating businesses and one related to the services possibly offered by the LBSN provider.

Monetary Losses: LBSNs have recently evolved into an *inexpensive* marketing channel for local businesses. Users can obtain special offers by checking-in at participating venues. This gives the latter an opportunity to be advertised, in an inexpensive way, only to people that actually have the potential to visit them. A traditional advertisement, which targets the majority of the population is expensive because of the vol-

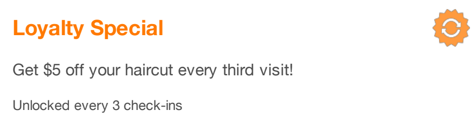


Figure 1. An example of a special offer requiring 3 check-ins.

ume of its target mass. However, only a percentage of the people exposed to it can actually benefit from the advertised good/service (e.g., people that are spatially located nearby).

Special deals lead to a temporary loss for the locale offering it. This is especially true for one-time deals, such as the ones offered in Groupon [5]. The rational behind these offers is that people visiting the venue will come back and hence, this will make up for the temporary loss. Hence, participating venues in LBSN offers might require more than one visit in an attempt to minimize the associated loss. For instance, Figure 1 provides an example of a special offer, which requires 3 check-ins. If the cost of the offer is c (in our example $c = \$5$), the locale's gain would be reduced by c for every visit if the deal was offered to every check-in. By requiring three check-ins the gain is only reduced by $\frac{c}{3}$ for every visit.

Jack who wants to unlock this offer but does not want to have to go three times to the venue and spend on average $s - \frac{c}{3}$ per visit, where s is the average expenditure of a client in the locale, can game the system and create two initial fake check-ins. This will enable him be present in the locale only once and unlock the deal. His expenditure will be only $s - c$, and the venue owner will have a reduced gain (the cost of the offer will again be c per visit). Assuming that $s = \$20$ in our example, the locale's gain is reduced by \$5/client instead of the $\frac{5}{3} = \$1.6$/client that was the target, while the cost per visit for Jack is reduced to \$15 instead of $20 - \frac{5}{3} = \$18.4$. Hence, it is evident that there can be monetary losses for businesses that want to use LBSNs as an advertisement channel. A detection system should be developed in order to filter forged check-ins and establish a secure way for venues offering deals. If the latter is not in place, business owners will have a reduced incentive to participate in similar systems.

Degraded Services: Recently, Foursquare - the largest LBSN to date - launched a novel recommendation engine, which considers check-ins from all users in order to provide recommendations [2] [3]. This engine takes into account user's check-ins, friend's check-ins, venue's check-ins and many other factors in order to provide suggestions. It should be evident that *noisy* data will not yield high quality service. Therefore, not only should LBSNs filter fake check-ins that can harm businesses (as aforementioned), but also identify any kind of fake check-ins (e.g., from gamers that simply want to gain as many virtual rewards as possible by checking-in to venues they have never been). Unless only "True" data are used from the LBSN provider to provide services, the latter will be degraded and of low quality.

Hence, it is evident that fake check-in detection is crucial to the long-run success of the LBSN paradigm. In the following section, we present our initial efforts on this problem.

FAKE CHECK-IN DETECTION

Cheating Model: In our work we consider two types of fake check-ins; (i) users can modify their GPS API and check-in a venue that is located far away and (ii) users that check-in to a locale that is nearby even if they are not physically present in it. Note here, that approaches such as the cheater code would not be able to detect any of them. However, the latter would be able to detect users that do not alter their GPS API and check-in to a far away venue, and thus, we do not consider it in our work. A realistic assumption we make for this study is that the numbers of fake check-ins are less than true check-ins (assumption 1). In addition, true check-ins are spatially contained within the premises of a venue, while fake check-ins are distributed over a larger area outside the latter (assumption 2).

Detection Algorithm: To defend against fake check-ins, every mobile user needs to provide location evidence to the LBSN provider along with his check-in information. For issuing location evidence, the mobile device collects beacon frames sent by nearby WiFi APs and measures the received signal strength (RSS). This provides a vector $RSS = [rss_1\ rss_2\ ...\ rss_n]$, which combined with a vector containing the unique MAC addresses of each AP ($MAC = [mac_1\ mac_2\ ...\ mac_n]$) forms the location proof which is forwarded to the LBSN provider with the check-in.

For location verification of a check-in of user u at locale l, the LBSN provider utilizes the recent k proofs of users claiming presence in l. Then spatial clustering on the RSS space is performed using the density clustering algorithm DBSCAN [7] as described in what follows. Having a set of points (check-ins in our case) DBSCAN first calculates the neighborhood $N(p)$ for each point p. The latter consists of all points within distance ϵ from p (the distance is calculated over the RSS vectors). The algorithm proceeds by examining whether it can merge the neighborhood to an existing cluster. The latter is possible if the neighborhood shares at least one common point to a cluster. Otherwise, if $|N(p)| \geq \texttt{MinPts}$ a new cluster is created. However, if $|N(p)| < \texttt{MinPts}$, p with its neighborhood are considered "noise". Figure 2 depicts the high level approach of

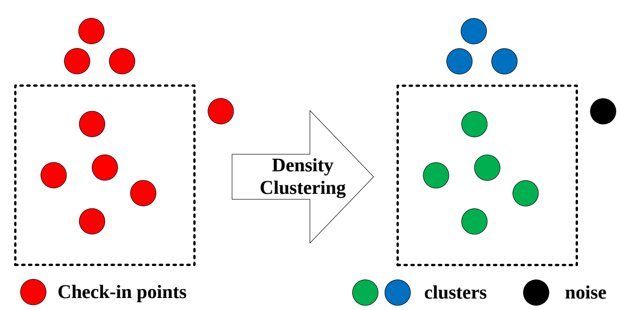


Figure 2. Pictorial representation of spatial clustering (`MinPits`=3).

DBSCAN. Clearly, ϵ and `MinPts` are two parameters that dictate the clusters and "noise" points identified.

In our context, considering the points defined by the RSS vectors of the check-ins claimed in a specific venue, we expect the points originating within the venue to be closer, thus belonging to the same cluster, as compared to those created outside the locale, due to the different wireless signal propagation path. In other words, real and fake check-ins will not be clustered together. Furthermore, we expect the fake check-ins to form clusters of lower cardinality (assumption 1) and/or be "noise" points (assumption 2).

The LBSN provider keeps track of the check-ins to a specific venue l and utilizes the latest k of them. Initially, when there are less than k prior check-ins none of them is classified. Once k of them are obtained spatial clustering is applied. Based on our discussion above, the cluster that includes the most points is flagged as "True" check-ins. The rest points (i.e., check-ins) are classified as "Fake".

Let us now consider Jack claiming to be in locale l. Using a sliding window approach, the k latest classified check-ins (say set S) together with that of Jack form the input to the density clustering. There are two possibilities for Jack's check-in; either classified as "noise" or belong to a cluster. In the former case the check-in is regarded as "Fake" (assumption 2). In the latter case, if the corresponding point belongs to the cluster with the largest cardinality then the check-in is flagged as "True", otherwise as "Fake" (assumption 1).

Note here that, while assumptions 1 and 2 might hold in the long run and over the total check-in set, it might be the case that they do not hold true for a subset of them (i.e., a specific set S). In order to avoid cascades of misclassification, every time a new check-in at l *arrives* we perform a reclustering. However, note that we only decide for the latest check-in in time; there is currently no feedback control for reinforcing previous check-ins classification.

One could use all the check-in history of a locale in order to apply the density clustering. However, evidence and RSS vectors might become stale due to the temporal variations of the wireless channel, as well as changes in the WiFi deployments. Exactly these are the factors that can provide robustness of our approach to replay attacks, where users record the location proofs at time t_1 and provide them with a fake check-in at time t_2.

Simulation and Evaluation: To evaluate our design, we simulate the check-in process over a virtual grid of locales. Venues are grouped into blocks of 6 and arranged in a 2D plane separated by streets. Venues within a block are tangent and separated by walls. 90% of venues are equipped with a WiFi access point. Our simulations include 24 venues (i.e., 4 blocks) and 20 users.

LBSN users follow the RANK model [12] to decide the next destination to check in. According to this model, the probability a user check in venue $v \in U$ from original venue $u \in U$ is defined as:

$$P_{uv} = \frac{rank_u(v)^{-\alpha}}{\sum\limits_{u \in U} rank_u(v)^{-\alpha}} \tag{1}$$

where,

$$rank_u\left(v\right) = \left|\left\{w \in U : d\left(u, w\right) < d\left(u, v\right)\right\}\right| \tag{2}$$

$d(u, w)$ is the distance between locales u and w. We have also used $\alpha = 0.84$ [12]. For a user who is truthfully checking-in to locale l, his actually position within l is randomly chosen. A user who performs a fake check-in, will be positioned randomly outside the venue, where the probability density of the distance follows an exponential distribution.

We also use a wireless signal propagation model for the RSS values recorded from the users. In particular, we use the Attenuation Factor Model [15]:

$$RSS = P\left(d_0\right) + 10n\log\left(\frac{d}{d_0}\right) + n_w \cdot W + v, \tag{3}$$

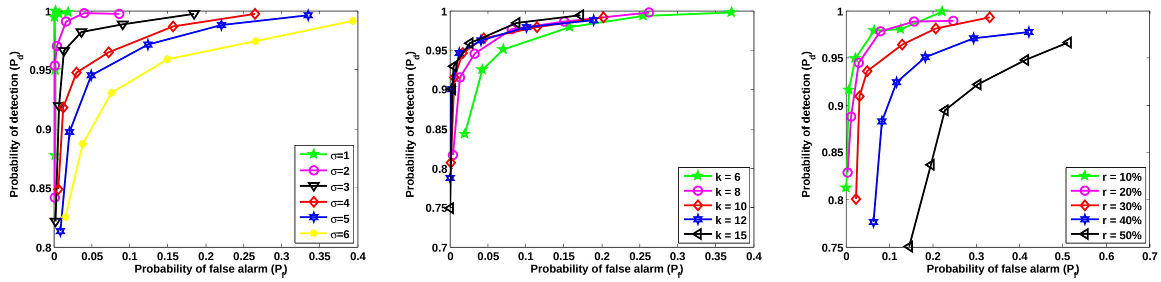
where, $P(d_0)$ (dBm) is the signal strength at distance d_0, n is the path loss exponent, W is an wall attenuation factor and n_w is the number of obstacles along the direct signal propagation path between transmitter and receiver, and v is Gaussian with $N \sim \left(u, \sigma^2\right)$. In our simulations, we set $d_0 = 1m$, $P(d_0) = -30dBm$, $n = -2.4$, $W = -15dBm$ and $u = 0$. σ is varied as described below.

We evaluate the performance of our fake check-in detection scheme with regards to the variations of the wireless channel as captured by the deviation σ, the percentage r of actual fake check-ins present in the system and different window sizes k. We set `MinPts`=3. ϵ determines the tradeoff between the probability of detection (a "Fake" check-in correctly classified) and false alarm (a "True" check-in classified as "Fake") and is used as the parameter for obtaining our results.

Figure 3(a) depicts the detection and false alarm probabilities for different σ ($k = 8$, $r = 20\%$). Each point in the curve is obtained for a different ϵ. As we can see the performance is much better when the wireless channel is stable. However, in a stable environment replay attacks can be more successfully since location proofs do not change over time. Nevertheless, even in a highly variable environment ($\sigma = 6$), the algorithms performs efficiently and is also more robust to replay attacks.

Figure 3(b) presents the performance of our scheme for different k ($\sigma = 4$ and $r = 20\%$). Again each point on the curves is obtained for a different threshold ϵ. As one might have expected the more points we input to the clustering algorithm, the more accurate detection it can perform.

Finally, Figure 3(c) presents the results for a varying percentage of fake check-ins r ($\sigma = 4$, $k = 8$). We can see that



(a) Variations in the wireless environment can degrade performance.

(b) The longer the history, the better the performance of our scheme.

(c) Detection works best when the number of fake check-ins is small.

Figure 3. ROC curves for our detection scheme.

the system performs better when r is small. When the latter increases, density clustering performance can be degraded, especially during the initialization phase, where the cluster including the most points is considered as the "True".

In all of the above results as we increase ϵ we move from the top right of the curve to the bottom left. Larger ϵ translates to higher probability that a point can be connected to a cluster. This, decreases the detection probability since it is easier for a fake check-in to fall into a "True" cluster, but also decreases the probability of false alarm.

Future directions: In the above we have presented our initial approach towards identifying fake check-ins. We seek to further extend our approach by examining other clustering algorithms and implementing a prototype system that would enable evaluation in a real environment. In particular, we want to examine: (i) possibilities for feedback control as aforementioned and (ii) the robustness of our approach to replay attacks. Finally, while the assumption of fake users being less than the real users is realistic, we opt to investigate different approaches whose performance is not affected by the proportion of the fake users.

CONCLUSIONS

In this work we have studied the problem of fake check-in information in LBSNs. We have argued for the importance and cruciality of this issue by analyzing possible effects from counterfeit spatial information. We have further designed and evaluate via simulations a detection system combining clustering algorithms and primitives of location proofs. We believe that our study will raise the awareness of the LBSN community and stimulate further research on the topic.

REFERENCES

1. Foursquare's cheater code: http://blog.foursquare.com/2010/04/07/503822143/.
2. Foursquare's recommendation engine: http://engineering.foursquare.com/2011/08/03/foursquares-data-and-the-explore-recommendation-engine/.
3. Foursquare's redesign: http://techcrunch.com/2012/06/03/foursquare-redesign-coming/.
4. Foursquare's special offers: https://foursquare.com/business/merchants/specials.
5. J. Byers, M. Mitzenmacher, and G. Zervas. Daily deals: prediction, social diffusion, and reputational ramifications. In *WSDM*, 2012.
6. D. E. Denning and P. F. Macdoran. Location-Based Authentication : Grounding Cyberspace for Better Security. *Computer Fraud and Security*, (February):12–16, 1996.
7. M. Ester, X. Xu, H.-P. Kriegel, and J. Sander. *Density-based algorithm for discovering clusters in large spatial databases with noise*, pages 226–231. AAAI, 1996.
8. W. He, X. Liu, and M. Ren. Location cheating: A security challenge to location-based social network services. In *IEEE ICDCS*, 2011.
9. M. Kjaergaard, M. Wirz, D. Roggen, and G. Troster. Mobile sensing of pedestrian flocks in indoor environments using wifi signals. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 95 –102, march 2012.
10. D. Lian and X. Xie. Learning location naming from user check-in histories. In *ACM SIGSPATIAL GIS*, 2011.
11. J. Lindqvist, J. Cranshaw, J. Wiese, J. Hong, and J. Zimmerman. I'm the mayor of my house: Examining why people use foursquare - a social-driven location sharing application. In *ACM CHI*, 2011.
12. A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo. A tale of many cities: universal patters in human urban mobility. In *PLoS ONE 7(5): e37027. doi:10.1371/journal.pone.0037027*, 2012.
13. K. Pelechrinis, V. Zadorozhny, and V. Oleshchuk. Collaborative assessment of information provider's reliability and expertise using subjective logic. In *CollaborateCom*, 2011.
14. S. Saroiu and A. Wolman. Enabling new mobile applications with location proofs. *Proceedings of the 10th workshop on Mobile Computing Systems and Applications - HotMobile '09*, pages 1–6, 2009.

15. S. Seidel and T. Rappaport. 914 mhz path loss prediction models for indoor wireless communications in multifloored buildings. *Antennas and Propagation, IEEE Transactions on*, 40(2):207–217, 1992.

16. J. Zhang, M. A. Ackerman, and L. Adamic. Expertise networks in online communities: Structure and algorithms. In *WWW*, 2007.

17. Y. Zhang, Z. Li, and W. Trappe. Power-modulated challenge-response schemes for verifying location claims. *IEEE GLOBECOM*, pages 39–43, 2007.

18. Y. Zheng and X. Zhou. *Computing with Spatial Trajectories*. Springer-Verlag New York, LLC, 2011.