mathematics

Article

Mobile User Location Inference Attacks Fusing with Multiple Background Knowledge in Location-Based Social Networks

Xiao Pan, Weizhang Chen and Lei Wu *

School of Economics and Management, Shijiazhuang Tiedao University, Shijiazhuang 050043, China; smallpx@stdu.edu.cn (X.P.); wzhchen1358@stdu.edu.cn (W.C.)
* Correspondence: wulei@stdu.edu.cn

Received: 28 December 2019; Accepted: 13 February 2020; Published: 17 February 2020

Abstract: Location-based social networks have been widely used. However, due to the lack of effective and safe data management, a large number of privacy disclosures commonly occur. Thus, academia and industry have needed to focus more on location privacy protection. This paper proposes a novel location attack method using multiple background options to infer the hidden locations of mobile users. In order to estimate the possibility of a hidden position being visited by a user, two hidden location attack models are proposed, i.e., a Bayesian hidden location inference model and the multi-factor fusion based hidden location inference model. Multiple background factors, including the check-in sequences, temporal information, user social networks, personalized service preferences, point of interest (POI) popularities, etc., are considered in the two models. Moreover, a hidden location inference algorithm is provided as well. Finally, a series of experiments are conducted on two real check-in data examples to evaluate the accuracy of the model and verify the validity of the proposed algorithm. The experimental results show that multiple background knowledge fusion provides benefits for improving location inference precision.

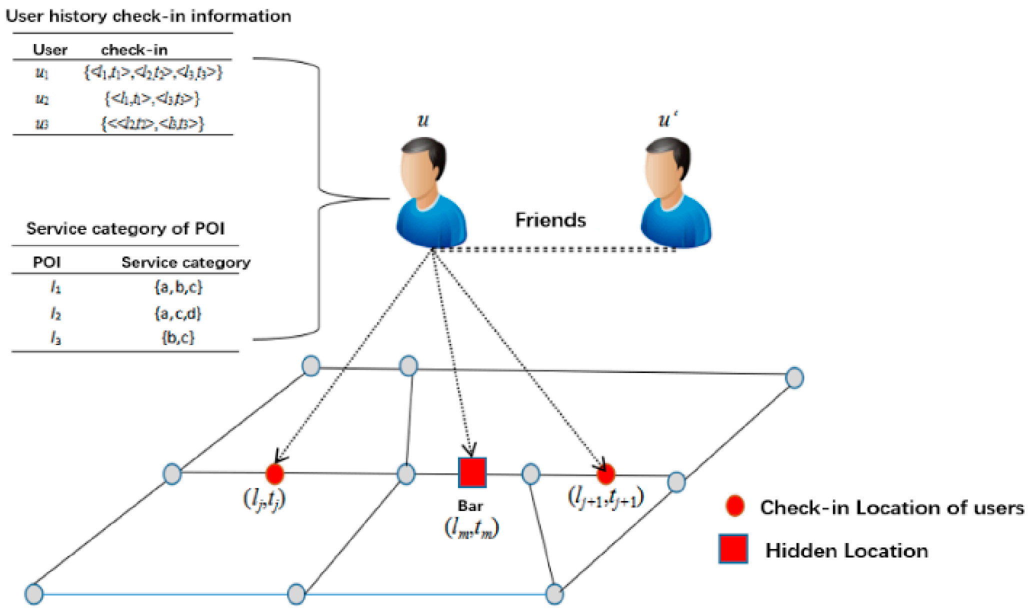
Keywords: hidden location inference; location privacy; location-based social networks; mobile computing

1. Introduction

With the rapid development of mobile intelligent terminal and communication equipment, Location based Social Networks (LBSN) have become popular, which put information and friends in easy reach. Users can use LBSN to share their locations (usually referred to as a “check-in”) with their friends, leave comments, and find coupons [1,2]. Many of these services have already been adopted and used by millions of users, and the number keeps growing steadily. However, due to the lack of effective and safe data management, users’ privacy is under the risk of leakage. Many research efforts have been devoted to investigating location privacy protection [3–6].

The simplest method to protect a user’s location privacy in LBSN is to make sure that users do not check in to the places which they regard as being sensitive (e.g., churches, hospitals, or bars) after they visit them. However, adversaries can still infer the hidden locations where a user has been visited without check-ins through linkages of multiple background information. Reference [7] refers to this kind of attacks as hidden location inference attacks.

Here, we use an example to illustrate hidden location inference attacks. As shown in Figure 1, a user u moves to the location l_{j+1} after he/she checks-in to the location l_j on the road network. During the path from l_j to l_{j+1}, user u visited a bar l_m on this path. In fact, u has not checked-in at l_m due to privacy protection concerns. However, if the attacker knows that u is a nightclub enthusiast, the attacker can infer that u would visit the bar l_m with a high probability when u checks-in at the location l_{j+1} [7]. u expects to avoid checking-in at any sensitive locations. However, u cannot know whether his/her future check-in behavior will lead to the disclosure of his/her hidden locations. There are two challenges for resolving this problem. One is how to find out if there is potential hidden location leakage with a suitable check-in precision while retaining a good user experience; the other challenge is how to measure diverse background knowledge. Reference [7] proposed a privacy alert mechanism to protect against hidden location inference attacks. However, it only employs geographical and social information, and much diverse background knowledge is not considered, which limits the inference accuracy.



User history check-in information

User	check-in
u_1	$\{<l_1,t_1>,<l_2,t_2>,<l_3,t_3>\}$
u_2	$\{<l_1,t_1>,<l_3,t_3>\}$
u_3	$\{<<l_2,t_2>,<l_3,t_3>\}$

Service category of POI

POI	Service category
l_1	{a,b,c}
l_2	{a,c,d}
l_3	{b,c}

Figure 1. Illustration for hidden location attacks.

After analyzing the LBSN service, we observed that attackers can collect background knowledge from two areas: the users' behavior characteristics (i.e., historical check-in sequence, personalized point of interests (POIs) preferences, and social networks) and the features of the physical world (i.e., geographical locations and POIs popularity). We elaborate on each background knowledge area one by one:

(1) The historical check-in sequence. The check-in sequence is constituted by the user's check-in locations sorted by the timestamps. Obviously, the check-in sequences reflect the user's behavior. For example, from the historical check-in records, the number of check-in sequences including the positions l_j and l_{j+1} both is 10. Then, we find that the user checks-in at l_m 3 times between l_j and l_{j+1}. Thus, we guess that the user will check-in to the position l_m from the location l_j to the location l_{j+1} with the probability 0.3(=3/10) in the future.

(2) Personalized POIs preferences. In LBSN, each POI is labeled with several service categories. Generally, users' preferences for the service categories are different. For example, nightclub enthusiasts prefer to visit different bars, while travel enthusiasts like to visit different tourist attractions. Therefore, the personalized preference of the service category can also be used to infer the probability of the user visiting the hidden location l_m.

(3) Social networks. Generally, the behaviors of friends are similar. Specifically, users often wander around various streets with their friends and go to a good restaurant or a shopping mall, etc. A user is more likely go to a place recommended by his/her friends. Suppose that a user u's friend u* always checks-in to the l_m. Then, although u does not check-in at l_m during the movement from

l_j to l_{j+1}, the attackers can also infer the likelihood that u visited l_m based on the user similarities between u and u*.

(4) Geographical location. In general, the geographical proximity of POIs has a significant impact on the user's check-in behavior. The probability of accessing l_{i+1} after checking-in at l_i depends on the distance between the two POIs. For example, users usually go to a mall or a movie theater nearby for convenience. On the other hand, the reachability of a position can be used to prune a sensitive hidden location. Specifically, if a user takes a short traveling time that is less than the minimum time required between the two locations l_i and l_{j+1}, then the user u certainly cannot visit l_m. That is, the location l_m is not reachable.

(5) POIs Popularity. If a POI is prevalent, then the POI is more attractive to a user. That means, the visiting probability of the users to this POI will be high. Therefore, we can use the popularity of POIs to infer the accessing probability as well.

Employing the above background knowledge, we propose two hidden location inference attacks models, namely WBI (weighted Bayesian hidden location inference model) and HLPI (hidden location prediction inference model through multi-factor fusion), to infer the hidden location of a user. WBI infers the access probability of the hidden locations based on the weighted Bayesian model, which considers the user's historical check-in sequences. The user's social networks, the personalized POI preference, and the POI popularity are used as the affected weights in WBI. HLPI is a hidden location inference model with multi-information fusion. The prior probability of the hidden location is computed based on the geographic location and the social network, and the POI popularity, respectively. A posterior probability is calculated according to the location reachability. Based on the two inference models, a hidden location inference attack algorithm is further given. According to the user's current check-in location l_{j+1} and the previous check-in location l_j, the probability of visiting the hidden location by the user is inferred. The most probable leaked hidden locations are pushed to users. We use a mechanism of advanced alerts to remind the user that his/her privacy is in danger. The users could make a decision according to their privacy preferences.

2. Literature Review

In the early stage, to protect location privacy, Gruteser and Grunwald [8] proposed a location k-anonymity model which was developed from the well-known k-anonymity concept, that is, the location is made indistinguishable by using the location information of at least $k - 1$ other users. Then, a great of research efforts [9–13] have been devoted in investigating location privacy protection. The existing location privacy protection mechanisms can be classified into a priori protection and a posterior screening [14]. In a priori protection mechanism, an actual location is replaced with an obfuscated location before the location is released. The mainstream idea for location obfuscation includes generalization, cryptography, generating dummies, and adding noise. In contrast, in a posterior screening mechanism, the user's location is protected from a service result perspective, assuming that a service request is answered. Our work focuses on the location attack method to infer the hidden locations of mobile users, and employs a warning mechanism to protect privacy. Thus, our work falls into the category of a posterior screening.

Privacy is an important issue when users consider using LBSN. Thus, location privacy in LBSN keeps attracting more attention from both academia and industry. Reference [15,16] focuses on helping management privacy, where a user can set location sharing privacy preferences. Reference [17] introduced a machine learning approach to control the sharing policy. From the attack models aspect, absent privacy attacks, nearby friends' attacks, and dynamic location inference attacks are studied widely.

Reference [18] is the first work that studied absent privacy protection model in LBSN. Absence privacy is a special kind of location privacy, which means that an attacker can know that a user is not in a position during a period of time. Reference [18] proposed an algorithm WYSE (Watch Your Social stEp) based on the spatial and temporal generalization. However, since this method needs to postpone releasing the generalized location, the quality of service (QoS) decreases. In order to protect from the absent privacy problem, Reference [19] proposed a POI-based absent privacy protection algorithm.

Nearby friends services as one of important types of LBSN can lead to user location disclose as well [14]. Reference [14,20] proposed a location privacy protection method to protect against these nearby friends attacks. Meanwhile, two location privacy protection algorithms based on symmetric encryption algorithm are proposed [20]. Reference [21] proposed a method to infer the location of users in LBSN, which employs the friends' location and attribute information. A dynamic Bayesian network model is used to calculate the user's access probabilities and trained by a real check-in data set.

Dynamic location inference attacks include hidden location inference, target location inference attack, and continuous location attacks. The most related work to ours is Reference [7], where a hidden location inference attack model is proposed. Reference [7] puts forward four inference algorithms, which are based on the simple Bayesian inference, collaborative filtering algorithm, and Markov inference model respectively. Compared with Reference [7], the method proposed in this paper fuses more diverse background knowledge, including a location check-in sequence, temporal factors, user social networks, personalized service preference, and the popularity of POIs. Reference [22] proposed a customizable and continuous privacy-preserving check-in data publishing framework through obfuscating user check-in data. The protection mechanism between reference [22] and our work is different. Reference [23] proposed a novel destination prediction attack and corresponding location privacy protection method. However, the research targets between reference [23] and our work are different. Hidden location inference aims to protect previously visited POIs in the past instead of prediction risk in the near future.

3. Background

Definition 1 (Check-in sequence). *A user's check-in sequence CS is a sequence of POIs sorted by timestamps. That is,* $CS = \{u_i, (l_1, t_1), \ldots, (l_j, t_j), \ldots, (l_n, t_n)\}$. u_i *is the user's identity.* $l_j = (x_j, y_j)$ *is the user's check-in location, which is a pair of longitude and latitude.* t_j *is the check-in timestamp.*

Definition 2 (Hidden Location). *Given a user's two check-in locations* l_j *and* l_{j+1} *at* t_j *and* t_{j+1} *respectively, the hidden position* l_m *is a POI being visited and not being checked-in at time* t_m $(t_j < t_m < t_{j+1})$ *on the path from* l_j *to* l_{j+1}.

Definition 3 (Check-in Matrix). *Given the historical check-in records of LBSN, we get a check-in matrix* $W_{|U|\times|L|}$, *where U and L denote the user set and POIs set, respectively. An entry* $w_{u,l}$ *in* $W_{|U|\times|L|}$ *is the frequency of the user u checks-in at location* $l \in L$.

Definition 4 (Social Matrix). *A user's social matrix* $F_{|U|\times|U|}$ *can be obtained from LBSN. If u and u′ are friends, the entry* $f_{u,u'}$ *is 1, otherwise* $f_{u,u'}$ *is 0.*

In fact, the social matrix is sparse, since most of the elements in $F_{|U|\times|U|}$ are zero. If two users share more common friends and more common check-in locations, the more similar the two users are. As a result, the two users are more likely to visit the same place [24]. Therefore, the user similarity is defined from the view of the user's check-in behavior and their social networks.

Definition 5 (User Similarity Matrix). *Given the check-in matrix* $W_{|U|\times|L|}$ *and the social matrix* $F_{|U|\times|U|}$, $s_{u,u'}$ *is an entry in the user's similarity matrix* $S_{|U|\times|U|}$, *which is defined as follows:*

$$s_{u,u'} = \theta\left(\frac{F_u \cap F_{u'}}{F_u \cup F_{u'}}\right) + (1-\theta)\cdot\left(\frac{L_u \cap L_{u'}}{L_u \cup L_{u'}}\right) \tag{1}$$

where θ is the system parameter between [0,1], F_u represents the friend set of user u, and L_u represents the check-in location sequences of user u. Obviously, $0 \leq s_{u,u'} \leq 1$. The larger the entry $s_{u,u'}$ is, the more similar the two users are.

Definition 6 (Service Preference Matrix). *Given check-in records and POI categories, the data entry $c_{u,c}$ in the user's service preference matrix $C_{|U|\times|C|}$ is the category frequency, i.e., how many times the user u visits a c(∈C)-type categorical POI [25]. Specifically, $c_{u,c}$ equals the number of c categorical POI visited by u divided by the total check-in number of u. It is worth noting that a POI can be labeled by using multiple categories.*

Example 1. *Figure 1 shows three users' check-in records, and each POI is associated with several service categories. The user u_1 checks-in at location {l_1, l_2, l_3}. According to the service categories of POI, l_1 is labeled by {a, b, c}, l_2 is label with {a, c, d} and l_3 is labeled with {b, c}. Thus, as Definition 6, the categorical preference for user u_1 is {$\frac{2}{3}$, $\frac{2}{3}$,1, $\frac{1}{3}$} for the service category {a, b, c, d,} respectively.*

Definition 7 (Popularity Matrix). *The popularity matrix is denoted as $P_{|C|\times|L|}$. The element $p_{c,l}$ in $P_{|C|\times|L|}$ represents the popularity of a POI in the view of c category. Specifically, $p_{c,l}$ is the number of the l POI check-in over the total check-in times for all POIs with the label c [25].*

Example 2. *Continuing with the above example in Figure 1, l_1 and l_2 are both labeled with the category a. From Figure 1, the category a has been checked-in four times. That is, l_1 is checked-in two times and l_2 is checked-in two times, respectively. Thus, the popularity of l_1 with the service category a is 1/2. In the same way, the popularity of l_1 with other service categories (i.e., b, c, and d) can be calculated. Thus, the popularity of l_1 with the service categories {a, b, c, d} is {1/2, 2/5, 2/7, 0}, respectively.*

4. Hidden Location Inference Attack Models and Algorithm

4.1. WBI: Weighted Bayesian Hidden Location Inference Model

The simple Bayesian model can be used to derive the visiting probability of a hidden location from the user's historical check-in records. Given a hidden location $\mathrm{l_m}$ and the upper bound of the time difference Δt between two check-in locations, the probability for the user visiting the hidden location $\mathrm{l_m}$ can be calculated as [7]:

$$\mathrm{P}_{u,l} = \frac{\sum_h C_h^{j,m,j+1} \cdot P(\Delta s \leq \Delta t)}{\sum_h C_h^{j,j+1}}. \tag{2}$$

In the formula (2), Δs is the check-in time difference between the two locations $\mathrm{l_j}$ and $\mathrm{l_{j+1}}$. $C_h^{j,j+1}$ indicates whether a check-in sequence h contains the both locations $\mathrm{l_j}$ and $\mathrm{l_{j+1}}$. If the answer is yes, $C_h^{j,j+1} = 1$, otherwise $C_h^{j,j+1} = 0$. Similarly, $C_h^{j,m,j+1} = 1$ indicates that a check-in sequence contains the position $\mathrm{l_j}$, $\mathrm{l_m}$ and $\mathrm{l_{j+1}}$ simultaneously; otherwise, $C_h^{j,m,j+1} = 0$.

We observe many factors can affect the user's visiting behavior, including personalized service preferences, POI popularity, and social networks. Intuitively, a popular POI is more attractive to a user. Thus, the probability of users accessing a popular location could be high. In addition, Reference [7] verified that friends have an important influence on each other's behaviors. Therefore, we propose a weighted Bayesian hidden location inference model WBI. The affecting factors are used as the weights in the simple Bayesian model. As a result, the user similarity, the personalized service preference, and the POI popularity are merged together to infer the hidden location visiting probability:

$$\mathrm{P}_{u,l} = \frac{\sum_h (1 + s_{u,u'} + c_{u,c} + p_{c,l}) \cdot C_h^{j,m,j+1} \cdot P(\Delta s \leq \Delta t)}{\sum_h (1 + s_{u,u'} + c_{u,c} + p_{c,l}) \cdot C_h^{j,j+1}} \tag{3}$$

In the formula (3), u′ and u are friends. $s_{u,u'}$ is the user similarity between u′ and u, which is calculated by Definition 5. $c_{u,c}$ is the service preference of the user u on the category c, which is calculated by Definition 6. $p_{c,l}$ is the POI popularity and is calculated by using Definition 7.

4.2. Hlpi: Hidden Location Inference Model Based on Multi-Factor Fusion

The probability of a user visiting a hidden location is influenced by various of background knowledge. Finding a proper model to fuse these background knowledges together is a challenge task. The existing work about location recommendation inspires us. Inspired by References [25,26], we propose a hidden location inference model HLPI. The basic idea is as follows. We first calculate the prior probability of visiting the hidden location $\mathrm{l_m}$ employing the association of geo-location, the social association of the user and the popularity and category of the POI. Then, the user's posterior probability of visiting the hidden location is computed using the prior probability and the reachability between two locations. The posterior probability indicates the likelihood that a user will visit the hidden location.

In general, the geo-location proximity between POIs has a significant impact on users' check-in behavior [25]. Two closer POIs are more likely to be visited consecutively than the ones that are far away. For example, a user usually goes to a nearby movie theater after shopping in a mall. Therefore, the visiting probability $\mathrm{P(x_{u,l})}$ for the hidden locations can be calculated by analyzing the geographical association between the checked-in POIs and the hidden locations.

In real life, the user's social relationship also has an impact on the user's behavior. For example, a user is likely to visit a POI (such as restaurants, shopping malls, etc.) which is recommended by his/her friends. That is, if two friends share more common checked-in POIs, they are more likely to visit the same location. We use the user's social relationship as an another factor to infer the probability $\mathrm{P(y_{u,l})}$ of a user visiting a hidden location l.

In addition, personalized preferences and the popularity of POI categories also have an influence on user's checked-in behavior. For example, nightclub enthusiasts will often hang out in different bars, and traveling enthusiasts will more likely travel to different tourist attractions. Meanwhile, more popular POIs are more attractive. Therefore, the personalized preference for the service category and the popularity of POI categories are also used to infer the visiting probability $\mathrm{P(z_{u,l})}$ of a hidden location l.

Reference [25] proves that the above three factors are independent of each other and follow identically distributed principles. Therefore, we integrate the factors together by multiplying the three probabilities. The prior probability of the user visiting the hiding location l_m is calculated as formula (4):

$$P_{u,l}^{prior} = P(x_{u,l}) \cdot P(y_{u,l}) \cdot P(z_{u,l}). \tag{4}$$

When a user checks-in at positions $\mathrm{l_j}$, $\mathrm{l_{j+1}}$ consecutively, the posterior probability of the user visiting the hidden location l_m is calculated as follows:

$$P_{u,l}^{posterior} = P(C_u^{j,m,j+1}|\Delta s) = \frac{P_{u,l}^{prior} \cdot P(\Delta s \le \Delta t)}{P(\Delta s)}. \tag{5}$$

Combining formula (4) and formula (5), the posterior probability of visiting a hidden location is given in formula (6):

$$\mathrm{P}_{u,l} = \sqrt[3]{P_{u,l}^{prior}} \cdot P(\Delta s \le \Delta t) = \sqrt[3]{P(x_{u,l}) \cdot P(y_{u,l}) \cdot P(z_{u,l})} \cdot P(\Delta s \le \Delta t) \tag{6}$$

It is notable that in order to balance the effect of the prior probability and the temporal probability, the cube root of the prior probability is used.

Specifically, P $\mathrm{(x_{u,l})}$ is estimated using the adaptive kernel estimation method. The ideas for computing $\mathrm{P(y_{u,l})}$ and P $\mathrm{(z_{u,l})}$ are as follows. We first get a probability density functions through the frequency distributions of the check-in frequency under the influence of user association and the category popularity, respectively. Then, the probability density functions are used to calculate the cumulative distribution so as to obtain the corresponding probability estimation [25].

4.3. Hidden Location Inference Attack Algoriyhm

In order to protect against hidden location leakage, we precede the check-in operation with a warning message, indicating that the user's privacy is protected. We employ the client-authorization server-LBSN server system architecture [7]. The specific system procedure is as follows:

(1) The sensitive category set SS_{u} which the user u regards to be sensitive is saved in the authorization server. The authorization server is trusted. When the user u wants to use the check-in services, u can send the check-in request with pre-check-in location $\mathrm{l}_{\mathrm{j+1}}$ at time $\mathrm{t}_{\mathrm{j+1}}$ to the authorization server.

(2) When the authorization server receives users' check-in requests, Algorithm 1 is utilized. The hidden locations between the user's previous check-in location lj and pre-check-in location $\mathrm{l}_{\mathrm{j+1}}$ are computed. The inferred hidden POIs are sorted by the computed visiting probabilities. The authorization server will send a privacy warning message to u when the categories of the hidden POIs fall into the sensitive category set SS_{u}. The most probable POIs whose category is sensitive are pushed to u in the warning message. The warning message will ask whether the use still wants to check-in at POI $\mathrm{l}_{\mathrm{j+1}}$ at time $\mathrm{t}_{\mathrm{j+1}}$.

(3) The users can make a choice by themselves. If the user still wants to check-in at location lj, the authorization server will forward the check-in request to the LBS server. Otherwise, the authorization server will drop this check-in request, meaning the check-in service is sacrificed while the user's privacy is protected.

Algorithm 1 shows the algorithm of hidden location inference. First, given the user's pre-check-in location $\mathrm{l}_{\mathrm{j+1}}$ and the previous check-in location l_{j}, the shortest path and popular path $\{\mathrm{SP}^2\}$ between the two locations are computed on the road network. We employ the methods in Reference [7] getting the set of hidden locations $\{\mathrm{L}_{\mathrm{m}}\}$ from $\{\mathrm{SP}^2\}$. If the WBI attack model is applied, we use the formula (3) to calculate the visiting probability for each position l_{j} in $\{\mathrm{L}_{\mathrm{m}}\}$. If the HLPI attack model is used, the formula (6) is used for computing the visiting probability. The matrices used in the formula (3) and the formula (6) are initialized through aggregating the users' historical check-in data. Finally, a pair set $\{<\mathrm{l}_{\mathrm{m}}, \mathrm{p}_{\mathrm{m}}>\}$ of the hidden location and accessing probability is returned.

Algorithm 1. Hidden Location Inference Algorithm.

```
Input:   $l_j$, $l_{j+1}$, W, S, C, P, G=<V, E>
Output:  Pair set of hidden location and the probability <$L_m$, $P_m$>

{$SP^2$}←find the shortest path and popular paths between $l_j$ and $l_{j+1}$
{$L_m$}←find the probable hidden locations within $\Delta$t on {$SP^2$}
if WBI is selected then
    for each $l_j$ in {$L_m$} do
        $p_j$← formula (3)
        {<$L_m$, $P_m$>}←{<$L_m$, $P_m$>}∪<$l_j$, $p_j$>
    end for
Else
    for each $l_j$ in {$L_m$} do
        $p_j$←formula (6)
        {<$L_m$, $P_m$>}←{<$L_m$, $P_m$>}∪<$l_j$, $p_j$>
    end for
end if
return {<$L_m$, $P_m$>}
```

5. Results

5.1. Setting

We use two real datasets to verify the efficiency and the effectiveness of the two proposed models in our experiments. One is the check-in data from Foursquare [27] using the road network of New York. The other one is the check-in data from Yelp [25], using the Phoenix road network. Table 1 lists the statistics of the two data sets. We select 5000 users randomly. Since real hidden locations are invisible in check-in datasets, we make a similar assumption as in Reference [7], that a user visits a POI if and only if the checks in the POI. A hidden location dataset is generated as follows: given a user u_k, a hidden location set is generated by marking off l ($5 \leq l \leq 25$) POIs that u_k has checked in randomly. After the POIs are marked off, the time interval between two consecutive check-in POIs cannot be larger than five times the average time interval.

Table 1. Statistics of the two data sets.

Dataset	Foursquare	Yelp
Number of users	717,382	70,817
Number of point of interests (POIs)	49,027	15,579
Number of POI Category	602	591
Number of check-in	206,416	335,022
Number of friend pairs	2,767,235	303,032

The hidden location attack models and algorithms proposed in this paper are all implemented in MATLAB and run on a Windows 7 with a 2.4 GHz processor and 4 GB of memory. The proposed models WBI and HLPI are compared with WFI and CFI in Reference [7]. WFI is a Bayesian inference method that only takes the friend similarity of user into account. CFI is a user-based collaborative filtering inference model that infers the hidden locations of user using the user's check-in similarity.

5.2. Results and Analysis

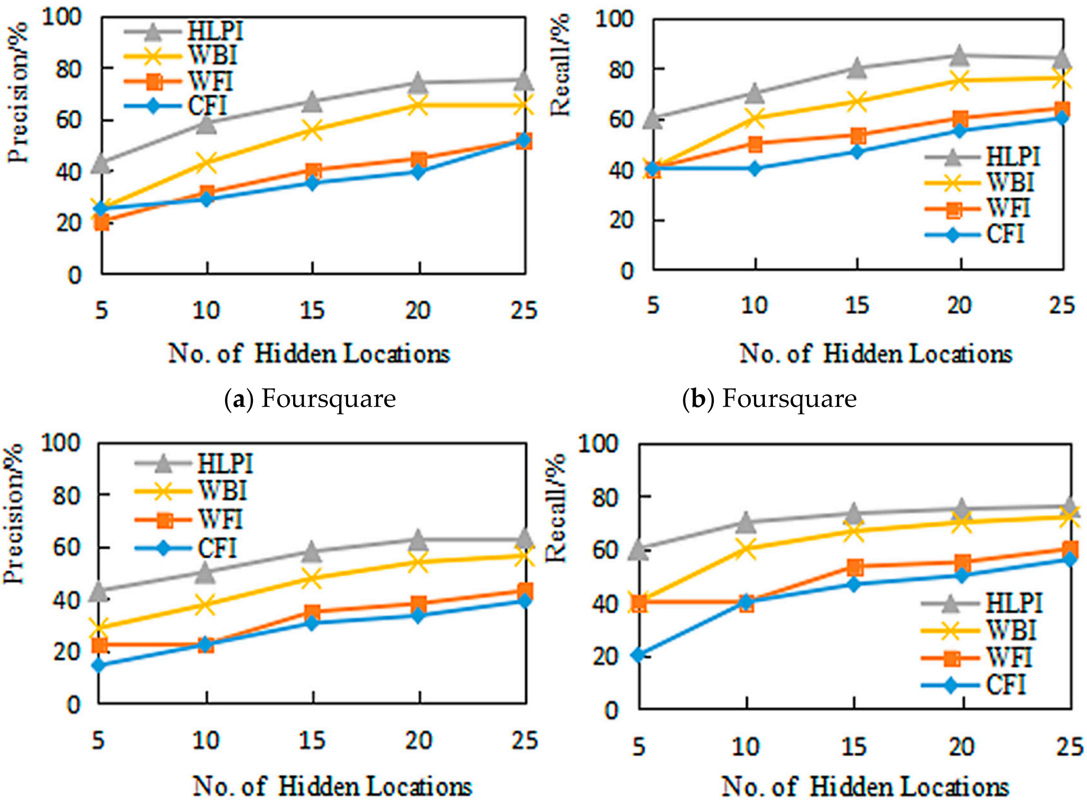
5.2.1. Accuracy

We evaluate the accuracy of the proposed infer models from two aspects (i.e., precision and recall) [28–32]. Precision refers to the percentage of the hidden locations generated by proposed models which are marked off POIs, and recall refers to the percentage of total marked off POIs correctly identified by the proposed models.

$$\mathit{Precision} = \frac{\text{number of true hidden locations returned from the model}}{\text{total number of locations returned from the model}} \tag{7}$$

$$\text{Recall} = \frac{\text{the number of hidden locations returned from the model}}{\text{the total number of marked off locations}} \tag{8}$$

Figure 2 shows the accuracy of the HLPI, WBI, WFI, and CFI with different numbers of hidden locations. As we can see from Figure 2, HLPI has the highest accuracy and recall among the four models no matter using the dataset from Foursquare or Yelp. That is because HLPI fuses more background knowledge than the other three models, including the user check-in records, the geographical location association, the user similarity, and the popularity of POIs. WBI comes second. The accuracy of WBI is higher than WFI, since WBI is improved from WFI by taking into consideration of the user category preferences and POI popularity. In the two check-in datasets, the precision and the recall of CFI are lowest with the existence of sparse matrices.

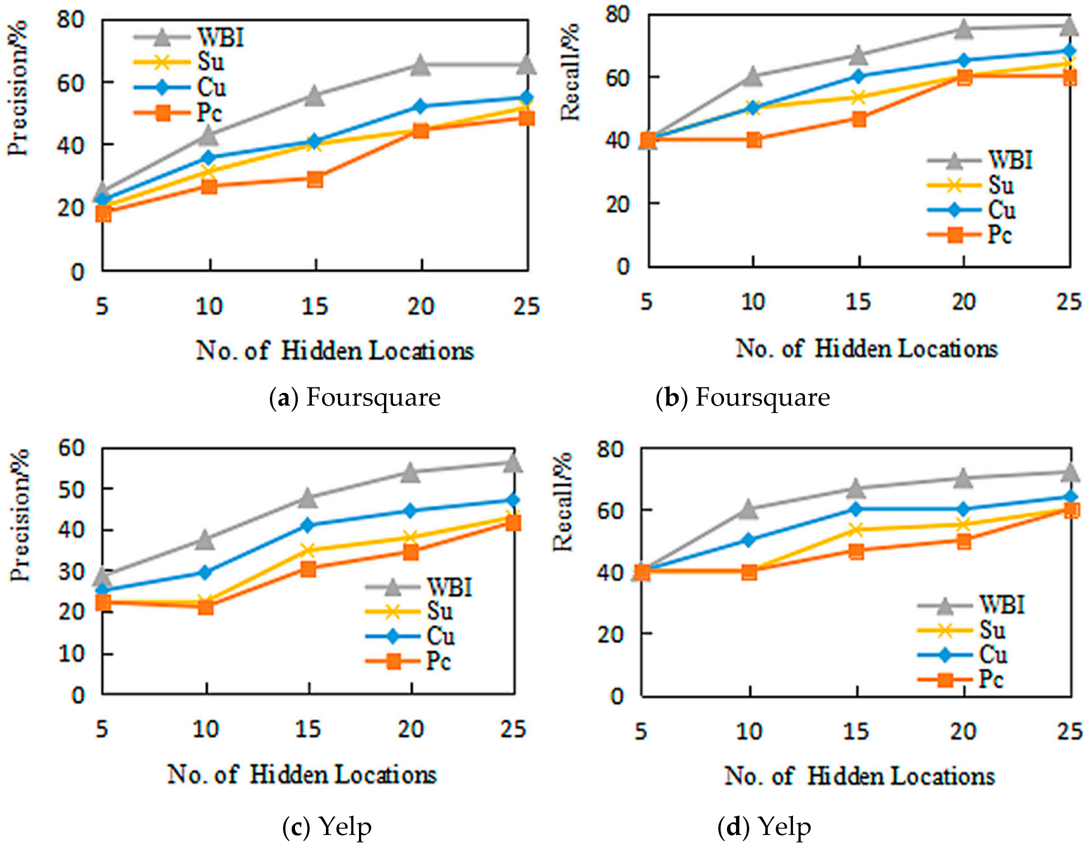


(**a**) Foursquare (**b**) Foursquare

(**c**) Yelp (**d**) Yelp

Figure 2. Prediction accuracy of four models.

Figure 3 shows the accuracy of WBI changes with different background knowledge under the different numbers of hidden locations. Specifically, we denoted WBI as Su when the friend similarity is only considered in the formula (3), is denoted as Cu when the user service category preference is only considered, and is denoted as Pc when the POI category is only considered. From Figure 3, the accuracy of the four methods increases with the increase of the number of hidden locations. When the three kinds of background are fused together, i.e., the accuracy of WBI model is highest. In addition, we can see that the accuracy of Cu is higher than that of Pc and Su in Figure 3, both in the Foursquare data set and the Yelp data set. This indicates that the user's personal service category preferences have the most impact on the probability of user visiting to hidden location. From Figure 3, the accuracy of WBI increases with the increase of the number of hidden locations. When the number of hidden locations is small, WBI is disturbed by users' un-common check-in behavior. The precision and recall are low. With the increase of the number of hidden locations, many true hidden locations are returned. Precision and recall increase both.

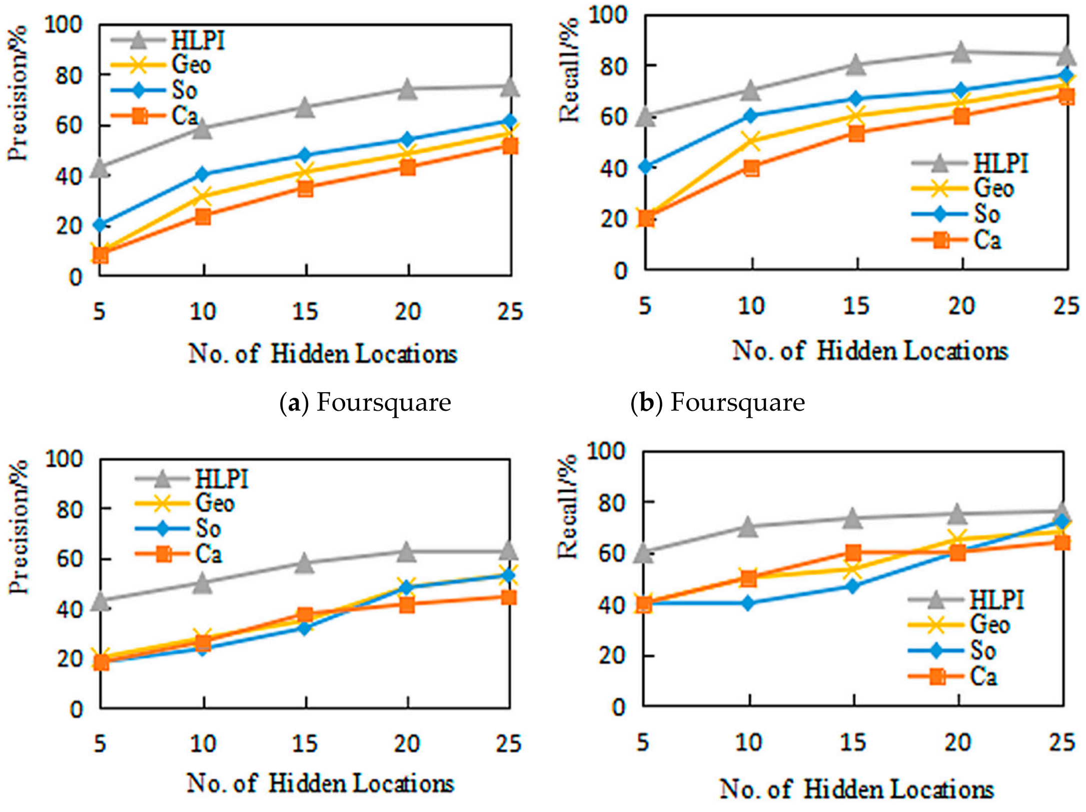


(**a**) Foursquare (**b**) Foursquare

(**c**) Yelp (**d**) Yelp

Figure 3. Prediction accuracy WBI with different weights.

Similarly, Figure 4 shows how the accuracy of the HLPI model changes with the number of hidden locations increasing when different background knowledge is considered. Geo only uses the association of geo-location in Equation (6); as a result, it only takes the user social association into account in Equation (6), and Ca only considers the POI categories' popularity in Equation (6). As shown in Figure 4, the accuracy of the four methods increases with the increase of the number of hidden locations. Moreover, the precision and recall of the HLPI, which fuses user social network and POI popularity together, are the best. Comparing the experimental results in the both datasets, it can be found that the influence of the user social network on the accuracy is more obvious than the geo-location association and the popularity of POIs in Foursquare. However, the opposite is the case on Yelp. That implies that user's check-in behavior is affected by various aspects, including user social network, geographic location, and POI category popularity association. In order to ensure prediction accuracy of hidden location inference, various backgrounds should be considered when calculating the visiting probability of a user.



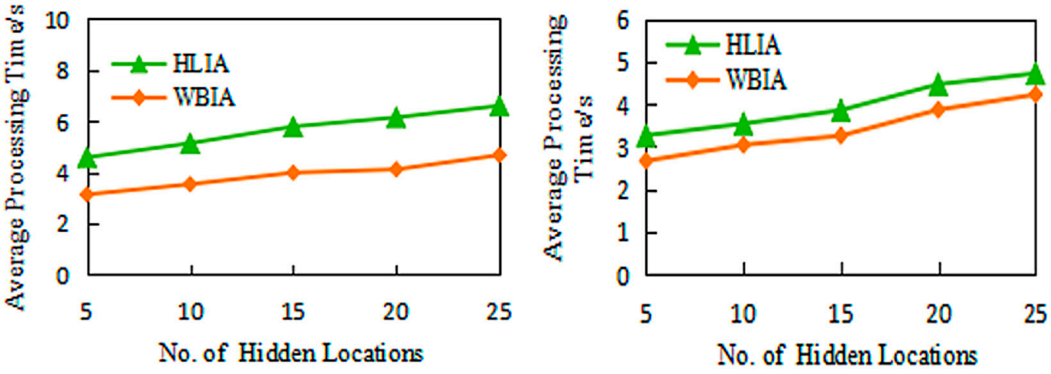
(**a**) Foursquare (**b**) Foursquare

(**c**) Yelp (**d**) Yelp

Figure 4. Prediction accuracy of HLPI with different types of background knowledge.

5.2.2. Effectiveness

This section evaluates the efficiency of Algorithm 1 under different numbers of hidden locations. The number of hidden locations increases from 5 to 25. Algorithm 1 is denoted as HLIA and WBIA when HLPI and WBI are used respectively. Figure 5 shows that the average processing time of the two algorithms increases with the increase of the number of hidden locations. In both datasets, the average processing time of the HLIA is higher than that of the WBIA. This is because HLPI needs to calculate the visiting probability from the geographical location, the social network of the user, the popularity of the POI, and the POI category, respectively. Then, the four aspects are fused together. However, WBIA needs the use of formula (3) only.



(**a**) Foursquare (**b**) Yelp

Figure 5. Efficiency.

6. Conclusions

Location-based social networks have been widely used, as a result the privacy leakage and protection raise more and more researchers' attention. Leakage of the hidden location threatens more dangerous to mobile users, since the users expect to hide these locations deliberately. This paper focuses on mobile user hidden location inference attacks when the attackers obtain various types of background knowledge. Considering location check-in records, reachability between to check-in locations, social networks, personalized service category preference, and POIs popularity, we propose two hidden location inference models and a hidden location inference attack algorithm. Finally, the accuracy of the models and the efficiency of the algorithm are evaluated using two real check-in datasets. The experimental results show that the prediction accuracy of the HLPI model is better than WBI, while the efficiency of HLPI is acceptable. In our current warning mechanism, users have to give up services when their privacy requirement is violated. Our future work will focus on developing a new protection method using a cryptography technique (e.g., geo-indistinguishability) in a new system architecture with more strong privacy protection and high service utility.

Author Contributions: X.P., W.C. and L.W. wrote the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Hebei Province of China (F2018210109), the Hebei Provincial Education Department Foundation (ZD2018040), Foundation of Introduction of Oversea Scholar (C201822), and the fourth Outstanding Youth Foundation of Shijiazhuang Tiedao University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vicente, C.R.; Freni, D.; Bettini, C.; Jensen, C.S. Location-Related Privacy in Geo-Social Networks. *IEEE Internet Comput.* **2011**, *15*, 20–27. [CrossRef]
2. Gao, H.; Liu, H. Data analysis on location-based social networks. In *Mobile Social Networking*; Springer: New York, NY, USA, 2013; pp. 165–194.
3. Cao, Y.; Xiao, Y.; Xiong, L.; Bai, L.; Yoshikawa, M. Protecting Spatiotemporal Event Privacy in Continuous Location-Based Services. *IEEE Trans. Knowl. Data Eng.* **2019**. [CrossRef]
4. Cao, Y.; Xiao, Y.; Xiong, L.; Bai, L. PriSTE: From Location Privacy to Spatiotemporal Event Privacy (short paper). In Proceedings of the 35th IEEE International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019.
5. Shao, Y.; Liu, J.; Shi, S.; Zhang, Y.; Cui, B. Fast De-anonymization of Social Networks with Structural Information. *J. Data Sci. Eng.* **2019**, *4*, 76–92. [CrossRef]
6. Singh, N.; Singh, A.K. Data Privacy Protection Mechanisms in Cloud. *J. Data Sci. Eng.* **2017**, *1*, 1–16. [CrossRef]
7. Huo, Z.; Meng, X.; Zhang, R. *Feel Free to Check-in: Privacy Alert against Hidden Location Inference Attacks in GeoSNs*; Springer: Berlin, Germany, 2013; pp. 377–391.
8. Gruteser, M.; Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In Proceedings of the International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, USA, 5–8 May 2003; pp. 31–42.
9. Andrés, M.E.; Bordenabe, N.E.; Chatzikokolakis, K.; Palamidessi, C. Geo-Indistinguishability: Differential privacy for location-based systems. In Proceedings of the ACM Conference on Computer and Communications Security, Berlin, Germany, 4–8 November 2013; pp. 901–914.
10. Mouratidis, K.; Yiu, M.L. Shortest path computation with no information leakage. *arXiv* **2012**, arXiv:1204.6076. [CrossRef]
11. Palanisamy, B.; Liu, L. Attack-resilient mix-zones over road networks: Architecture and algorithms. *Trans. Mob. Comput* **2015**, *14*, 495–508. [CrossRef]
12. Yao, L.; Wang, X.; Wang, X.; Hu, H.; Wu, G. Publishing Sensitive Trajectory Data Under Enhanced l-Diversity Model. In Proceedings of the 20th IEEE International Conference on Mobile Data Management, Hong Kong, China, 10–13 June 2019; pp. 160–169.

13. Ye, Q.; Hu, H.; Meng, X.; Zheng, H. PrivKV: Key-value data collection with local differential privacy. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 19–23 May 2019; pp. 317–331.
14. Pan, X.; Zhang, J.; Wang, F.; Philip, S.Y. DistSD: Distance-based social discovery with personalized posterior screening. In Proceedings of the IEEE International Conference on Big Data, Washington, DC, USA, 5–8 December 2016; pp. 1110–1119.
15. Toch, E.; Cranshaw, J.; Hankes-Drielsma, P.; Springfield, J.; Kelley, P.G.; Cranor, L.; Sadeh, N. Locaccino: A privacy-centric location sharing application. In Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing, Copenhagen, Denmark, 26–29 September 2010; pp. 381–382.
16. Sadeh, N.; Hong, J.; Cranor, L.; Fette, I.; Kelley, P.; Prabaker, M.; Rao, J. Understanding and capturing people's privacy policies in a mobile social networking application. *Pers. Ubiquitous Comput.* **2009**, *13*, 401–412. [CrossRef]
17. Kelley, P.G.; Hankes Drielsma, P.; Sadeh, N.; Cranor, L.F. Cranor User-controllable learning of security and privacy policies. In Proceedings of the 1st ACM workshop on Workshop on AISec, Melbourne, Australia, 19–20 August 2008; pp. 11–18.
18. Freni, D.; Ruiz Vicente, C.; Mascetti, S.; Bettini, C.; Jensen, C.S. Preserving location and absence privacy in geo-social networks. In Proceedings of the ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010; pp. 309–318.
19. Riboni, D.; Pareschi, L. Bettini Integrating Identity, Location, and Absence Privacy in Context-Aware Retrieval of Points of Interest. In Proceedings of the IEEE International Conference on Mobile Data Management, Lulea, Sweden, 6–9 June 2011; pp. 135–140.
20. Mascetti, S.; Freni, D.; Bettini, C.; Wang, X.S.; Jajodia, S. Privacy in geo-social networks: Proximity notification with untrusted service providers and curious buddies. *VLDB J.* **2011**, *20*, 541–566. [CrossRef]
21. Sadilek, A.; Kautz, H.; Bigham, J.P. Finding your friends and following them to where you are. In Proceedings of the ACM International Conference on Web Search and Data Mining, Seattle, WA, USA, 8–12 February 2012; pp. 723–732.
22. Yang, D.; Zhang, D.; Qu, B.; Cudré-Mauroux, P. PrivCheck: Privacy-Preserving Check-in Data Publishing for Personalized Location Based Services. In Proceedings of the Ubicomp, Heidelberg, Germany, 12–16 September 2016; pp. 545–556.
23. Xue, D.; Wu, L.F.; Li, H.B.; Hong, Z.; Zhou, Z.J. A novel destination prediction attack and corresponding location privacy protection method in geo-social networks. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1550147716685421. [CrossRef]
24. Ye, M.; Yin, P.; Lee, W.C.; Lee, D.L. Exploiting geographical influence for collaborative point-of-interest recommendation. In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 25–29 July 2011; pp. 325–334.
25. Zhang, J.D.; Chow, C.Y. GeoSoCa: Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations. In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 443–452.
26. Zhang, J.D.; Chow, C.Y. CoRe: Exploiting the personalized influence of two-dimensional geographic coordinates for location recommendations. *Inf. Sci.* **2015**, *293*, 163–181. [CrossRef]
27. Bao, J.; Zheng, Y.; Mokbel, M.F. Location-based and preference-aware recommendation using sparse geo-social networking data. In Proceedings of the SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; pp. 199–208.
28. Hu, L.; Sun, A.; Liu, Y. Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction. In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 6–11 July 2014; pp. 345–354.
29. Yin, H.; Cui, B.; Sun, Y.; Hu, Z.; Chen, L. LCARS: A spatial item recommender system. *ACM Trans. Inf. Syst.* **2014**, *32*, 1–37. [CrossRef]
30. Yin, H.; Sun, Y.; Cui, B.; Hu, Z.; Chen, L. LCARS: A location-content-aware recommender system. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 221–229.

31. Ying, J.J.C.; Kuo, W.N.; Tseng, V.S.; Lu, E.H.C. Mining User Check-In Behavior with a Random Walk for Urban Point-of-Interest Recommendations. *ACM Trans Intell. Syst. Technol.* **2014**, *5*, 1–26. [CrossRef]
32. Kosmides, P.; Demestichas, K.; Adamopoulou, E.; Remoundou, C.; Loumiotis, I.; Theologou, M.; Anagnostou, M. Providing recommendations on location-based social networks. *J. Ambient Intell. Humaniz. Comput.* **2016**, *7*, 567–578. [CrossRef]