Wireless Personal Communications (2020) 114:949–973
https://doi.org/10.1007/s11277-020-07402-2

OMCPR: Optimal Mobility Aware Cache Data Pre-fetching and Replacement Policy Using Spatial K-Anonymity for LBS

Ajay K. Gupta[1] · **Udai Shanker**[1]

Published online: 23 April 2020

Abstract

Location-based services are an important category of context-aware computing, which play an important role in providing the continuous, local and spatially confined information systems very efficiently and accurately to their clients. The key component in location-based services is the current location of a mobile user. Thus, to protect mobile users' location and their other information to an untrusted party with consideration of minimal waiting time, Optimal Mobility Aware Cache data Pre-fetching and Replacement policy (OMCPR) is being proposed here. In this continuous location-based services model, the system introduces a mediator namely Anonymizer by employing the prefetching facility for spatial K-anonymity that resides in between the user and Query Analyser to form a cloaking region using mobile user inputs (data freshness, the contribution rate of cell's cache and location). It provides high-quality lossless location-based services by the utilization of the frequent pattern mining of mobile users' trajectories to forecast their next position as per mobility and multiple constraints. A client–server based queueing model is used to simulate the proposed OMCPR platform. It provides higher privacy protection than the current state of the art strategies available, also minimizes the overhead of the LBS server and waiting time of mobile users by the addition of the prefetching facilities to Anonymizer.

Keywords Location privacy · Caching · Prefetching · Mobility · Security · Authorization

1 Introduction

The key component in location-based services (LBS) [1] is the location; one's location gives information about what one is doing. For instance, he doesn't do the same exercises in his workplace as he does in his kitchen. The further prerequisite of current LBS is that they must "derive the purpose of the client" and reacts accordingly. Location dependent query is the continuous, local, and spatially confined query used in the LBS. Locations in a

✉ Ajay K. Gupta
ajay25g@gmail.com

Udai Shanker
udaigkp@gmail.com

[1] Department of Computer Science and Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur, India

location-dependent query are provided as a parameter implicitly using a global positioning system (GPS) or explicitly in the execution of a query. Passing the value of the parameter related to location implicitly using a GPS can be seen as local queries e.g. listing the local weather and locate the nearest restaurant. The explicit query can be seen as queries where the client gives the value to the location parameter e.g. find the nearest metro station to the passport office. Based on the context being captured in the system, LBS can be classified categorically into primary and secondary context system. The raw data, which can be acquired from location sensors, accelerometers, microphones, etc., are termed as primary context. This raw data may be deduced or refined to derive secondary context such as distance between targets, the direction of moving objects, velocity, etc. In many cases, other context information (e.g. audibility, pollution, temperature, etc.) is relevant to the given service. These contexts are closely related to the target location being considered in the LBS. To process the query in the requirement of the services responsible for the prediction of ahead of present route location, it is of utmost necessity for the consideration of location, movement direction & velocity of the users. The LBS face many inherent challenges such as low-quality communication, limited local resources, scarce bandwidth, and frequent network disconnections. Caching in LBS is used to cope up with some of the aforementioned shortcomings. In primitive LBS caching schemes, if a certain data object is not found in the mobile user's cache, then mobile users send this query with its location to Query Analyser. Query Analyser processes the received query and returns the result based on the client's location. The drawback of previous policies is that the Query Analyser is prone to leak the user's confidential information such as user location and associated data to an untrusted third party for business profits. Various policies exist to deal with the privacy challenges of the LBS. A trusted third party such as cloaking is popularly known previously used privacy policy for the LBS. In cloaking, the user location is anonymized by a circle. However, this degrades the point of interest (POI) service. Therefore, while implementing the LBS, one should consider these issues to enhance user privacy with the prior aim of improvement in the cache hit rate. To be wise and proactive, gadgets do not simply know the present context only rather they must have the capacity to anticipate future context also. System capacity to foresee future location is used for prefetching [2] in the cache. The prefetching function in any LBS minimizes the count of communication between mobile users and Query Analyser, and which in turn improves the user's privacy.

Whenever a client issues a query, it is first searched in the client cache, then in its neighbor cache and finally, in Anonymizer cache. If the queried result found in any one of the above caches, it is immediately returned to the mobile client; otherwise, the request is sent to the Anonymizer. Anonymizer, in turn, applies Frequent Pattern Mining of mobile user's trajectories for the user's next location prediction based on multiple constraints. It is then applied for the prefetching and spatial k-anonymity (PSKA) to form a cloaking region using mobile user inputs, i.e., data freshness, the contribution rate of the cell's cache and location in LBS. The cloaking region with the POI query is sent to the Query Analyzer. Query Analyzer does not obtain private information (like client exact location); rather, it requires the cloaking region and POI only. Anonymizer uses the cell's contribution rate in the cache, data freshness and also the user's predicted next location for cloaking region formulation. In this process, Anonymizer selects k cells satisfying k-anonymity criteria and chooses a cloaking region having the highest query probability among one of the k cells. The query probability for the cache jth data item in the cell_i is represented by P_i^j. Let the time elapsed in the last access to the jth data item be t_j^l, current system time be t_c. Suppose, the weightage related to the most recent access for the estimation of probability is represented by α, then update to the current access probability follows the equation given below.

$$P_i^j = \frac{\alpha}{t_c - t_i^l} + \frac{1-\alpha}{old\, P_i^j}$$

Query Analyser is aware of the cell's query probability P_i $(1 \le i \le M)$. However, as there are k cells and each cell has one or more users, the probability of being specifically responsible to send request would be at most 1/k. Prefetching and spatial K-anonymity (PSKA) techniques used in our policy cannot estimate the exact location of a user from the cloaking region received from Anonymizer. Thus, it is free from the inference attack of Query Analyzer.

1.1 Summary of Contributions

The major contributions of the paper are illustrated as follows.

(a) OMCPR provides a lossless service with the quality high and protects a user's location and his information to untrusted identity. A novel PSKA scheme used in OMCPR applies multi-level caching; moreover, privacy is improved as some of the user's query can be replied using cache without sending it to the untrusted Query Analyser.
(b) Anonymizer applies the Sequential Pattern Mining & Clustering-based user next location prediction model using the user's movement trajectory to predict the next location in the cache replacement and invalidation scheme.
(c) Anonymizer selects k number of cells satisfying k-anonymity criteria and processes these cells using parameters such as data freshness, cache contribution rate and predicted next location to get a cloaking region having the highest query probability among one of k cells.
(d) It analyses efficiency, overhead and privacy concerns of proposed OMCPR compared with the previously available schemes for the same.

1.2 Outline

The remaining portion of this paper is arranged as written ahead. The survey of the past works and finding out the problem statement of the proposed work have been carried out in Sect. 2. The preliminaries concepts including assumptions and LBS architecture description of the addressed domain are being discussed in Sect. 3. The suggested approach and main results of our studies are outlined in Sects. 4 and 5 respectively. Finally, the last Sect. 6 offers the conclusion of the paper and also enlists briefly the scopes for the research directions in LBS further.

2 Related Past Works and Problem Statement

Various schemes in LBS exist to protect location privacy. The existing approaches of protecting location can be categorized into 4 types, namely obfuscation, regulatory strategies, privacy policies, and anonymity. Most of the existing privacy works in LBS are based on obfuscation, location perturbation or anonymization. On the server side, the system is integrated with a trusted anonymization server [3]. At the client-side, some mobile device based schemes are integrated as proposed in the past [4–8]. Due to dependency on a trustworthy server, the former type struggles with a single point of failure e.g. location anonymizer in [9]. Therefore, a user's privacy will be violated if an attacker gains access to it. This trustworthy server is indeed a bottleneck of results because all the queries sent will move through it. The schemes based on mobile devices prevent these issues. Several works concentrate on VHC mapping [4], k-anonymous cloaking box [6] and find-based solution [5] to reduce the overhead related to storage and computation.

Unified grid and caching have been proposed by Zhang et al. [10] to enhance user privacy. Two users cannot query to the same region in this policy; due to this, the overhead of the client query became overwhelming in the system. The selection of two cache-based dummy locations has been added in the location privacy scheme by Niu et al. [11]. The drawback of this scheme is that it is not fit for continuous queries.

Apart from the above schemes, k-anonymity [12] is a popular scheme used as a user's location privacy policy. This scheme has been widely used in the previous LBS [13–16]. In the k-anonymity scheme, the system selects k neighboring cells satisfying k-anonymity criteria and chooses the cloaking region having the highest query probability among one of k cells. A potential client can't be differentiated from k-1 other clients by an attacker in the k-anonymity location privacy scheme. In LBS, there may exist a situation where a user does not want to participate in the user's location privacy scheme and also does not want to be part of the anonymity set. So, these cases should also care while designing the user's location privacy scheme in LBS.

Approaches such as location perturbation, spatial cloaking, and variants of spatial cloaking are widely studied techniques for preserving location information for snapshot-LBS. Approaches such as trajectory privacy using k-anonymity, privacy using fake trajectories or dummies and privacy through path confusion are popular techniques for preserving location information for continuous LBS. For continuous query scenario anonymization, either the online or historical trajectories of other users can be used. One of the limitations of the previous techniques was infrastructure dependency. A trusted middleware is required to compute the spatial cloaked region for the set of users requesting for the service in a period.

Another limitation of all existing k-anonymity based techniques is the inherently trusted assumption for all the users in anonymity set and believes to be reporting their real location. However, if the users are not trusted, they can inject fake location together with the fake query to the Anonymizer. Thus, it will increase the chances of performing location disclosure substantially higher than 1/k. This attack is named as location injection attack. Zhao et al. [17] have proposed a framework based on users' mobility similarity that does not require the knowledge about how fake locations are manipulated and shown to be effective by performing extensive simulations on real-world datasets.

In varieties of previously proposed spatial cloaking based privacy-preserving policy, the size of the cloaking region used can be much larger in areas having a low density of users. Density might be low either due to less usage of the services or being in a remote area. To handle the above limitation of previous policies, the spatial cloaking technique can be mixed with k-anonymity. Amini et al. [18] suggested enhancing consumer protection across the cache scheme. Cell users may query the POIs locally rather than submitting the requests to the untrustworthy. LBS server pre-fetches these data items within a specified region before reaching to that region. Mobile clients have to also archive massive volumes of data items for a broad region.

Shokri et al. [19] created a decentralized position confidentiality mechanism retaining a collaborative unit called MobiCrowd. The position exposing likelihood risk is reduced due to checking for utility details by users in nearby MobiCrowd unit before submitting a query to the

LBS server. Nevertheless, if users in the neighbor are not able to serve the result, the request will be sent to the LBS server and will, therefore, be at risk. It was not meant to increase the cache hit ratio that causes poor performance in caching. Mobicache [7] is attempting to store further data that is not yet stored and is not recognizing the side details that an attacker might have. Therefore, with the aid of side knowledge, the adversary may infer the actual position.

In the variants of previously proposed spatial cloaking based privacy-preserving policy, the size of the cloaking region used can be much larger in the areas having a low density of users. Size being inversely proportional to the quality of the service can degrade the quality or may sometimes end in denial of the services. Density might be low either due to less usage of the services or being in a remote area. To handle the above limitations of previous policies, the spatial cloaking technique can be mixed with k-anonymity. In all of the previous policies, there was always a trade-off between service quality and privacy. To satisfies the maximal query POI as requested by the mobile user, the LBS compromises user privacy. None of the previous LBS schemes considers the user mobility in LBS to serve the user's query. The overhead (e.g. CPU computation, memory, and battery energy) of the above policies to the system is high. Also, they don't even have a standardized privacy parameter to quantify the impact of cache on privacy. So, the caching architecture is relatively straightforward despite taking into consideration the significant factors e.g. data item freshness, query probability, etc. Therefore, the LBS should integrate an efficient caching scheme with better users' privacy and reduced server overhead. To deal with the above challenges, the aim here is to design a method adopting multilevel caching to improve user privacy in continuous LBS by the formulation of the cloaking region.

For the formulation of the cloaking region, the caching and prefetching system should integrate a next location prediction module with data freshness and cache contribution rate. Employing any prefetching scheme or next location prediction of the user in LBS can be seen as adding a recommendation system to the LBS. The various recommendation systems in LBS are described in this section. The real-world GPS data can be mined to get knowledge and answer two common questions. The first one is "where we shall go if we want to do something?". This corresponds to location recommendations. The second one is "what can we do in that place?". This corresponds to activity recommendations. The trajectory-based recommendation involves the finding of the similarities between people based on their mobility histories. To design recommendation systems, it is assumed that peoples having common mobility behaviors and mobility profiles are likely to be potential friends of each other and share preferences and interests. Hence, to visit the same types of places, they opt for the same type of route to reach the destination. Generally, trajectory-based recommendation applies frequent pattern mining to extract and compare location visit sequences. In this recommendation system, if any person gets similar users having common mobility behaviors and mobility profiles, they can be recommended to him or her as friends. Hence, they can be recommended to the individual as places to visit and routes to follow based on the place they visited and routes taken by them. The past strategies have utilized the moving user trajectories for location and activity recommendations. Some of the related works with trajectories based recommendation are listed in the forthcoming sub-sections.

2.1 Location Recommendation

In the early location recommendation system, the user's current location is used. In [20], a cyber-guide framework is created to provide librarian records that describe the close by homes and identification of related people. An item-based collaborative filtering method is used for developing a city voyager to recommend shops in [21]. It uses visited shop histories for a given item to recommend some shops to a user. In [22], Bayesian learning is employed to calculate recommendation values for different restaurants for providing a ranking list. It considers both the popular client locations and associated preferences to calculate recommendation values. A client-based collaborative selection policy [23] was proposed to suggest restaurants. Hypertext Induced Topic Search (HITS) based mobile user popular locations and access recommendation systems are proposed in [24], using past traveling and access histories. Li et al. [25] have proposed a place and friend suggestion framework which includes 3 sections namely users visited location history, similarity exploration of users and location recommendation. In the first section, a hierarchical graph is represented by the hierarchical clustering of stay points for each user using his location histories. In the second section, the user's similar sequence is explored by evaluating T-patterns [26]. This is achieved by retrieving a sequence of shared graph nodes between all users' graphs. User (node) similarity score is calculated from each other node and it is rated according to its level of similarities concerning a given user. The set of people (nodes) with higher similarity scores is recommended as potential friends to the user. In the third section, they have added the semantics of locations rather than geographic coordinates used for the locations with semantic meaning such as cinema hall, shopping mall, restaurant, etc.

2.2 Activity Recommendation

Making attention to future trends, the current research is migrated from the prediction of system network behavior to individual behavior prediction, i.e., from location prediction to context prediction or activity prediction. It is the latest research issue and very little researches have been done on it so far. The early research works are solely based on recognition of the user's movement from received GPS data by ubiquitous computing [27]. In [28], a hierarchal conditional random field model based on GPS data is used to recognize "where is a user?". In [29], everyday tasks such as teeth cleaning, driving, sleeping and cooking are recognized through the use of sensor namely RFID. The researchers integrate the additional sources of information also, i.e., context information or temporal information for developing an application for activity recognition and activity prediction. Through, market-based analysis, anyone can predict the interrelated activity which may happen together with a given activity. This can be used to find the relationship between places and actions e.g. client going for movie hall may go to the nearby cafeteria as well. In [30], Zheng et al. used collective matrix factorization through a collaborative filtering technique to train the model for location and activity recommendation. The future trend is to evolve more features related to the user to enhance the performance. The feature extraction involves the collection of raw input data and then pre-processes it to extract useful features. The various prediction algorithms exist, which can be used for prediction purposes by utilizing useful features. The features can be used to create a user's social network by which users of similar interests can share their experiences. The recommendation frameworks are useful as the likelihood subsequent locations and/or activities of mobile users inferred using the mobile users' trajectories can be integrated further in LBS to enhance the caching, prefetching and/or privacy.

3 System Architecture

This paper has proposed OMCPR model adopting multilevel caching to improve user privacy by the formulation of the cloaking region. Better cache hit ratio in turns results in better client privacy as well as reduces the LBS server overhead. OMCPR is consisting of the pre-fetching and spatial k-anonymity for better client privacy by forming a cloaking region using mobile user inputs (data freshness, the contribution rate of cell's cache and location) in continuous LBS. The above policy has utilized the Frequent Pattern Mining of mobile user trajectories for the user's next location prediction based on multiple constraints.

Here, the system introduces a mediator namely Anonymizer that resides in between the user and Query Analyser. Anonymizer hides the exact location of the mobile user. But, Anonymizer becomes the performance bottleneck of the overall system in this architecture because all the user's queries pass through the Anonymizer. Anonymizer selects k cells satisfying k-anonymity criteria and chooses the cloaking region having the highest query probability among one of the k cells. As there are k cells & each cell has one or more users, the probability of being a specific user who sends request would be at most 1/K. PSKA technique used in our policy protects the user's exact location by framing the cloaking region received from Anonymizer. Thus, it is free from the inference attack of Query Analyser. System architecture constitutes three entities namely Anonymizer, Query Analyser, and mobile clients. The main function of Anonymizer is to predict the next location of moving users, pre-fetch probable next query data and store received user's query results from Query Analyser into the cache. Anonymizer selects the cloaking region by spatial k-anonymity. Anonymizer randomly selects k neighboring cells satisfying k-anonymity criteria and chooses the cloaking region having the highest query probability among one of the k cells. Moreover, it has a history of issued queries from different users in a particular cell. Query Analyser is an LBS based application having map resources and location related points of interest (POI) such as nearest hospital, hotel, ATM, Saloon, Market, etc. It serves the location-based query associated with the respective cloaking region received from Anonymizer. The mobile client device has processing power, global positioning (e.g., GPS), data storage and ability to communicate with other neighboring mobile users. They work on a collaborative shared basis. It can share the stored cache query result. The LBS reference architecture, that has followed in proposed OMCPR, is depicted in Fig. 1. The internal architecture of Query Analyser is also shown in Fig. 2. Here, in this paper, LBS has used CELPB Invalidation, SPMC-CRP and PSKA for cache invalidation, cache replacement, and user's data privacy-preserving respectively.

3.1 Assumptions

Mobile clients and Anonymizer both are assumed that they report their right locations to mobile clients and Anonymizer is trusted in LBS. If initiating query mobile client reports fake location with a fake query to Anonymizer, then this phenomenon is known as location injection. For this scenario, the chances of privacy leakage of location and user's information would be more than 1/k. Within the client's communication range, none of the neighboring clients will send malicious information to the peer nodes. To protect the confidentiality and integrity of the transmitted data, the LBS has security schemes such as hashing and cryptographic schemes. The proposed model considers location privacy issues from the moving client's perspective. Moving clients can choose among privacy levels as strict,

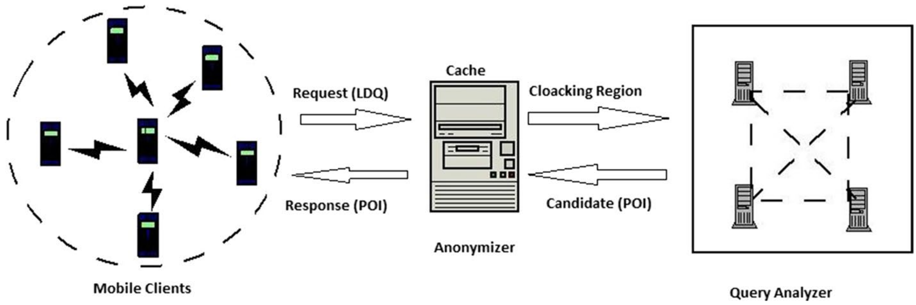


Fig. 1 Architecture of LBS

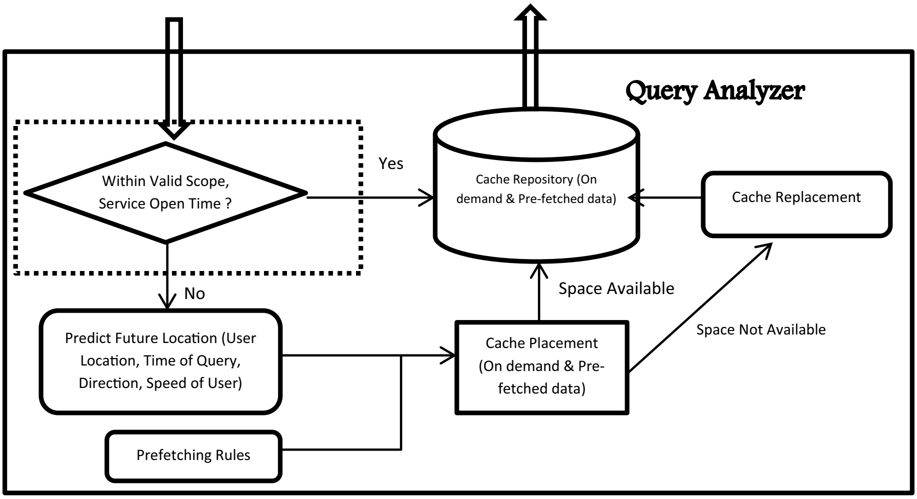

Query Analyzer
Within Valid Scope, Service Open Time ?
Yes
No
Cache Repository (On demand & Pre-fetched data)
Cache Replacement
Space Available
Space Not Available
Predict Future Location (User Location, Time of Query, Direction, Speed of User)
Cache Placement (On demand & Pre-fetched data)
Prefetching Rules

Fig. 2 Query analyzer

medium or low. Here, the desired threshold is being used based on service quality and privacy trade-off. If the threshold has a higher value, then LBS compromises user privacy and satisfies the maximal query POI as requested by a mobile user. The proposed OMCPR employed a pre-fetching and spatial K-anonymity (PSKA) scheme to improve the privacy scheme by forming the cloaking region in continuous LBS. To protect an individual's location and user's information to the untrusted party, a multi-level caching is proposed.

4 Proposed Optimal Mobility Aware Cache Data Pre-fetching and Replacement Policy (OMCPR)

Due to the mobility of users in LBS, it faces many inherent challenges such as low-quality communication, limited local resources, scarce bandwidth, and frequent network disconnections. Caching in LBS is used to cope up with some of the above shortcomings. To minimize the access cost and enhance the information accessibility, the system should cache the data items which are frequently queried on the mobile consumer side. In LBS, an invalidation mechanism ensures the consistency of data between the server and the application cache. The value of the queried data item (POI) by mobile client shows the variation in it when this data item is queried to access from different locations. The replacement procedure [31–33] is initiated whenever the cache becomes full and needs more data items to be stored in the cache. Our model has used SPMC-CRP [34] scheme as a cache replacement policy. Cache invalidation [35] is one of the important challenges in client cache management. If a cellular user keeps moving after submitting a query and has arrived at a new location until the response gets returned, the validity checking is desirable in this case. It may occur due to a long data access delay. Here, the term valid scope [36] is used to define the geometry shape or more precisely valid region, in which the value of a given Spatio-temporal variable is valid. CELPB invalidation scheme [37] is used in this paper for cache invalidation of a query result. Pre-fetching refers to fetching the probable future query result to mobile user cache before moving the user query. It improves the system performance by improving privacy, reduces the LBS server overhead and makes the system alive during disconnections. PSKA module is integrated with OMCPR for user's data pre-fetching and privacy preservation. PSKA takes inputs as various parameters like user query request time, service type, direction, speed, current location, services count, the semantic distance between the consecutive query, the spatial distance between service instances and access frequency to provide relevant pre-fetching detail. The PSKA policy can be extended by adding a PSO and/or fuzzy duration window with a threshold [38, 39] to decide the time after which the pre-fetched data would be expired. This is done because a mobile user might wait after getting service at a location before requesting to associate service at the same location. Depending on service types (e.g. hotel, restaurant, fuel station, etc.), the waiting time may vary. Deciding a time window helps in effective utilization of cache storage which in turn improves the cache hit ratio.

Here, POI information according to the mobile user's next location is stored at the user and Anonymizer caches it in advance. Some of the pre-fetching policies use moving user trajectories to access history to hoard the hot items lied on the path. The user and Anonymizer cache in our LBS have a tag bit to declare whether the stored cache data is on-demanded data or pre-fetched one. Query Analyser serves the location-based query associated with the respective cloaking region received from Anonymizer.

4.1 SPMC Based Next Location Estimation

Sequential Pattern Mining and Clustering technique involve five steps. The removal of outliers or noise (random movements) is the first step. In the second step, the minimum support threshold ($\mathrm{sup}_{\mathrm{min}}$) is used to discover all Frequent Mobility Patterns (FMP) from trajectory datasets (D). The trajectory dataset has different values for two attribute namely timestamp (T), and client location at this timestamp. All the same type of FMPs (L_k) are grouped into one cluster (C_i). Finally, n clusters C_1, C_2… C_n are formed with different sets of FMPs [24]. The third step is to find the best suitable cluster for the current moving client trajectory from different types of clusters. The fourth step involves the mining of FMPs and framing of mobility rules R. The mobility rules [25] are passed through the filter of confidence threshold (conf_min). In the final step, the matched rule is used to estimate the next cell_id in the predicted region.

This algorithm involves an iterative procedure to find user frequent mobility pattern set. In this procedure, all the Frequent Mobility 1-Patterns (i.e. k = 1), whose support value is less than the minimum support threshold, i.e., $supp_{min}$ are discarded from the

Table 1 Mobility trajectory log file

Sr. no.	Log file patterns
1	$<(l_5, t_3), (l_1, t_5), (l_{15}, t_7)>$
2	$<(l_5, t_3), (l_1, t_5), (l_{13}, t_9)>$
3	$<(l_5, t_3),(l_2, t_4), (l_1, t_5), (l_{15}, t_7)>$
4	$<(l_5, t_3),(l_2, t_4), (l_1, t_5), (l_{13}, t_9)>$
5	$<(l_2, t_4), (l_1, t_5), (l_{15}, t_7)>$
6	$<(l_2, t_4), (l_1, t_5), (l_{13}, t_9)>$
7	$<(l_3, t_2), (l_2, t_4), (l_{12}, t_6), (l_{13}, t_9)>$
8	$<(l_3, t_2), (l_2, t_4), (l_{12}, t_6), (l_{14}, t_8)>$
9	$<(l_4, t_1), (l_3, t_2), (l_2, t_4), (l_{12}, t_6), (l_{13}, t_9)>$
10	$<(l_4, t_1), (l_3, t_2), (l_2, t_4), (l_{12}, t_6), (l_{14}, t_8)>$

Table 2 Length-1 candidate pattern set and Length-1 frequent patterns set

*C*1		*L*1	
Length-1 candidate pattern set	Support	Length-1 frequent patterns set	Support
$<(l_5, t_3)>$	4	$<(l_5, t_3)>$	4
$<(l_2, t_4)>$	8	$<(l_2, t_4)>$	8
$<(l_3, t_2)>$	4	$<(l_3, t_2)>$	4
$<(l_4, t_1)>$	2	$<(l_1, t_5)>$	6
$<(l_1, t_5)>$	6	$<(l_{15}, t_7)>$	3
$<(l_{15}, t_7)>$	3	$<(l_{13}, t_9)>$	3
$<(l_{13}, t_9)>$	3	$<(l_{12}, t_6)>$	3
$<(l_{12}, t_6)>$	3		
$<(l_{14}, t_8)>$	2		

further computation. The remaining frequent Mobility 1-Patterns are processed for the next iterative step (i.e. k = 2). For k = 2, mobility patterns are termed as frequent Mobility 2-Patterns. The client mobility log file as given in Table 1 is used to understand the processing of frequent pattern mining. The dataset shows various locations with a respective timestamp of a day. Length-1 candidate pattern set and Length-1 frequent patterns set are given in Table 2. Considering this table, the support value of $<(l_4, t_1)>$, $<(l_{14}, t_8)>$ is less than $supp_{min} = 3$. So, they are discarded for subsequent computation. Length-2 candidate pattern set and Length-2 frequent patterns set are given in Table 3.

The same procedure is executed iteratively for each length-k patterns in L_k. $L_k = <(p_1, t_1), (p_2, t_2)\ldots (p_{k,} t_k)>$ is the Length-K large patterns set and Cell p_k neighboring cell set in coverage region is given by $N(p_k)$. C_k is known as candidate k-patterns. The next step in the algorithm is to find out all the suffix $s = (n, t_{k+1})$ from neighboring cell set satisfying the specified criteria as given below and attaching it at the end of Length-K large patterns set to generate a candidate $(k + 1)$-patterns.

$$U(p_k) \leftarrow \{s = (n, t_{k+1}) | < (n, t_{k+1}) > \in N(p_k) \text{ and } t_{k+1} > t_k\}$$

The set of all these suffix are represented by the set $U(p_k)$. The timestamp of p_k is t_k. Then, candidate (k + 1)-patterns selection is done by $supp_{min.}$

Table 3 Length-2 candidate pattern set and Length-2 frequent patterns set

C_2		$L2$	
Length-2 candidate pattern set	Support	Length-2 frequent patterns set	Support
<(l_5, t_3), (l_2, t_4)>	2	<(l_2, t_4), (l_{13}, t_9)>	4
<(l_5, t_3), (l_1, t_5)>	2	<(l_2, t_4), (l_{12}, t_6)>	4
<(l_5, t_3), (l_{13}, t_9)>	2	<(l_1, t_5), (l_{15}, t_7)>	3
<(l_2, t_4), (l_1, t_5)>	2	<(l_1, t_5), (l_{13}, t_9)>	3
<(l_2, t_4), (l_{13}, t_9)>	4		
<(l_2, t_4), (l_{12}, t_6)>	4		
<(l_3, t_2), (l_5, t_3)>	0		
<(l_3, t_2), (l_2, t_4)>	2		
<(l_1, t_5), (l_{15}, t_7)>	3		
<(l_1, t_5), (l_{13}, t_9)>	3		
<(l_{15}, t_7), (l_{13}, t_9)>	0		
<(l_{12}, t_6), (l_{13}, t_9)>	2		

Let, A and B are two FMPs. The mobility rules for given mobility patterns are as follows.

$$\mathrm{R}: \mathrm{A} \rightarrow \mathrm{B} \text{ such that } \mathrm{A} \cap \mathrm{B} = \emptyset.$$

Let, S is the set of all FMPs. The mobility rules are given below.

$$\mathbf{Z} \rightarrow (\mathbf{S} - \mathbf{Z}) \quad \text{for all } (\mathrm{Z} \subset \mathrm{S}) \text{ and } \left(\mathrm{Z} \neq \emptyset\right)$$

Algorithm A1: FMPs (L) Extraction

```
Input:   Mobility Log (D) with 1- patterns
         $Supp_{\min}$
         $L_1 \leftarrow$ Legth-1 large pattern set          // Let $L_k$ = Legth-$K$ large patterns set
         $C_1 \leftarrow$ Legth-$1$ candidate patterns set     // Let $C_k$ = Legth-$K$ candidate patterns set
         G = Directed graph for neighboring information,

Output:  $L$ = User FMP set.

Begin
    k = 0
Repeat
        k = k+1
        For each Legth-k-large patterns set $F_k = <(p_1, t_1), (p_2, t_2), \ldots, (p_k, t_k)> \in L_k$ do
                $N(p_k) \leftarrow \{n| n$ is the adjacent cell of $p_{k-1}\}$
                For each node $n \in N(l_{k-1})$ do
                        $U(p_k) \leftarrow \{s=(n, t_k)| <(n, t_k)> \in N(p_{k-1})$ and $t_{k+1} > t_k\}$
                        For each $s=(n, t_k) \in U(p_k)$ do
                        $C=<(p_1, t_1), (p_2, t_2), \ldots, (p_k, t_k), (n, t_{k+1})>$
                        $C_{k+1} = C_{k+1} \cup C$
                        End for
                End for
        End for
        For each mobility pattern $B \in D$ do
                $F \leftarrow \{q | q \in C_{k+1}$ and q is a subsequence of B$\}$
                For each $c \in C$ do
                        q.count = q.count + 1
                End for
        End for
        $L_{k+1} \leftarrow \{q | q \in C_{k+1}$ and $c$.support $\geq$ $supp_{min}\}$
        $L = L \cup L_{k+1}$
While $L_k \neq \emptyset$
Return $L$
End
```

Algorithm A2: Mobility Rules Formulation

```
Input:    L: User FMPs set.
          conf_min : Minimum confidence threshold.
Output:   Rules : frequent mobility rules set.
Begin
  Rules ← $\phi$
  For each User FMPs $M_k$ = <($p_1$, $t_1$), ($p_2$, $t_2$), …, ($p_k$, $t_k$)> ∈ $L_k$ , k>=2
      $M_l$ ← $M_k$
      y=k
      Repeat
        //A ← (y-1) sub pattern of $M_l$ .
        A ← <($p_1$, $t_1$), ($p_2$, $t_2$), …, ($p_{y-1}$, $t_{y-1}$)>.
        conf = support.($P_k$)/support.(A).
        If conf ≥ conf_min then.
              R ← {<($p_1$, $t_1$), ($p_2$, $t_2$), …, ($p_{y-1}$, $t_{y-1}$)>→<($c_l$, $t_l$), …, ($c_k$, $t_k$)>}        //R ← (A → $P_k$-A).
              R.w = (RuleTime-MinTime) / (MaxTime-MinTime)×100.
              Rules = Rules ∪ R.
        Else  break.
        End if.
        y = y –1.
      While y >1.
  End for.
      Return Rules.
End
```

Every rule (R) will have its corresponding confidence value, i.e., confidence(R).

$$\text{Confidence}\,(\text{R}) = \frac{support\,(A \cup B)}{support\,(A)} \times 100$$

This Policy uses a confidence threshold to filter the various frequent mobility rules. The most important rules are those generated by the use of the most recent mobility patterns. Each rule has its associated temporal weighted value. The mobility rules are generated for a user near places using frequent patterns mining. A temporal weighted value (w_i) is assigned to each rule (r_i). Those rules are more important, which are generated using recent mobility patterns. Let, the time of the last point in rules tail is given by $\text{Rule}_{\text{Time}}$. Max_{Time} and Min_{Time} are the recorded maximum and minimum time in log file respectively. The weighted value is calculated by the following equation.

$$weight\,(R) = \frac{Rule_{Time} - Min_{Time}}{Max_{Time} - Min_{Time}} * 100$$

Algorithm A3: Prediction_Next_Location(P, R)

```
Input:    Rules Set = R
          Prediction Count Threshold = m
          Moving client's trajectory P = <($l_1$, $t_1$),( $l_2$, $t_2$),( $l_3$, $t_3$),……,( $l_{n-1}$, $t_{n-1}$)>.
Output:   Next_Cell_ID $L_p$
Begin
    $L_p = \phi$
    j =1
    For each rule r in R:<($r_1$,$t_1$'),( $r_2$,$t_2$'),....,( $r_k$,$t_k$')> → <($r_{k+1}$,$t_{k+1}$'),…,( $r_t$,$t_t$')>.
        If <($r_1$,$t_1$'),( $r_2$,$t_2$'),..,( $r_k$,$t_k$')> is contained by P = <($l_1$,$t_1$),( $l_2$,$t_2$),( $l_3$,$t_3$),..,( $l_{n-1}$, $t_{n-1}$)>and $r_k = l_{n-1}$.
            $T_{diff}$ = 1/(|$t_{n-1}$ – $t_k$'|+1).
            r.matchingscore = r.w + $T_{diff}$.
            r.score= r.matchingscore + r.support + r.confidence.
            MatchingRules = MatchingRules ∪ r .
            NextCells[j] = (r.score, $a_{k+1}$)
            j = j +1.
        End if
    End for
    Descending order Sorting of NextCells array.
    i = 0
    Select the starting x TuplesArray elements
    While (i < x and i < TuppleArray.length) do
    $L_p \leftarrow L_p \cup$ TuppleArray[i]
    Cell_ID = $L_p$
    Return Cell_ID
End
```

4.2 Forming Cloaking Region

By using predicted next location with data freshness and cache contribution rate, Anonymizer selects k cells satisfying k-anonymity criteria and chooses the cloaking region having the highest query probability among one of the k cells. The system has an M × M grid rectangular service area where each cell is of equal size having a unique identifier (c_i, r_j), where, c_i is the column id and r_j is row id. The system follows a cooperative caching scheme for user's cache sharing similar to that used in [40]. In this architecture, both the mobile clients and Anonymizer is assumed to be inherently trusted as shown in Fig. 3a and reports their real locations. If fake users are present in the system as shown in Fig. 3b, they can inject fake locations together with the fake query to the Anonymizer. Thus, this increases the chances of performing location disclosure substantially higher than 1/k.

In the simulation, the point's density within the cell shows the likelihood that the given moving client can issue a query within that cell. The cells showed blank means that it has no user's queries in it. This area may be mountains, lakes or swamps thereon. The mobile user submitting a query is present somewhere in cells. While forming a cloaking region, the system involves a procedure to find some cells around moving users that have the highest query probabilities and total different background knowledge. In this way, the attacker can not infer the related background knowledge from a group of participating cells in the cloaking region.

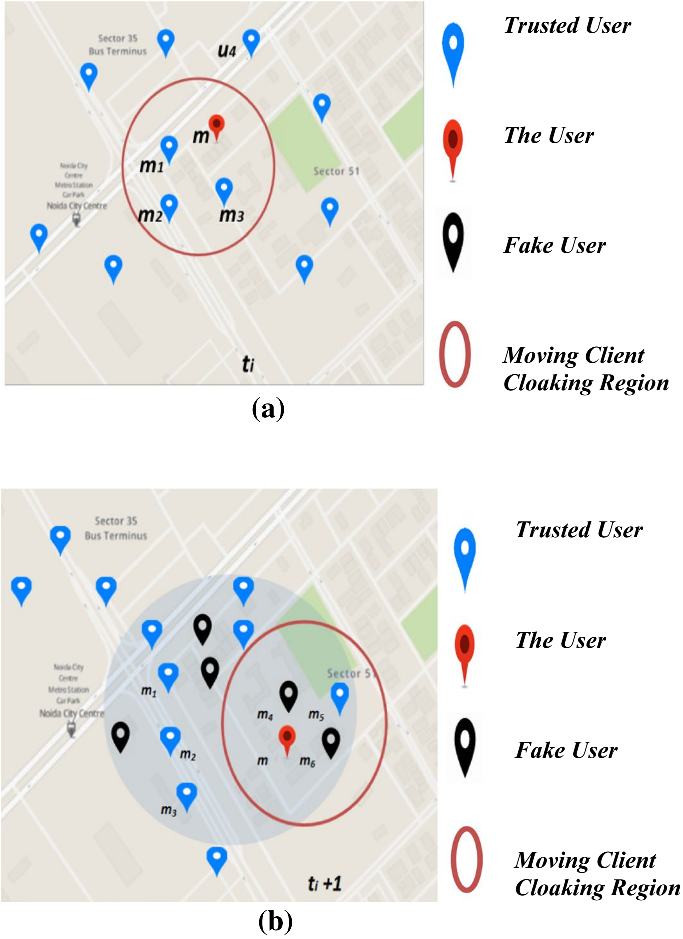


Fig. 3 Continuous Queries for 4-Anonymity **a** without Location Injection Attack, **b** with Location Injection Attack

In system architecture, let I_q is the set of all the cell identifiers which can contain the query result. I_c is the matched cell identifier in the user's cache. I_n is the matched cell identifier from the user's neighboring cache. L_u indicates the user's location who initiates the query. D_u depicts moving client movement direction. Theta (θ) is the desired threshold ($0 < \theta \leq 1$) based on service quality and privacy trade-off. If θ has a higher value, then it means that LBS has compromised user privacy and satisfied the maximal query POI as requested by the mobile user. The minimum number of cell identifiers required to query by Anonymizer is represented by λ and $\lambda = \theta * \mathrm{Num}(I_q) - \mathrm{Num}(I_c) - \mathrm{Num}(I_n)$, where Num() is the number of cell identifiers function. The OMCPR internal process of pre-fetching and spatial K-anonymity (PSKA) follows the following steps.

1. First of all, the system finds cell identifiers I_q within a radius (R) of the query range. After that, the user searches his cache for identical POIs for these identifiers I_q. Here, in this case, $I_u = I_c$ is the matched cell identifiers for the user's cache contribution.

2. Whenever, the user issue a query, it is first searched into its cache. If the queried data items are not found in the user's cache, then its neighboring user's cache is searched. If the queried result found in any of them, it is immediately returned to the requesting mobile user. Here, in this case, ($I_u^1 = I_q - I_c$) is the set of cell identifiers that need to be queried for the user's neighbor cache contribution.
3. If the queried data items are not found in the user's neighboring cache, then the mobile user needs to send a message of inquiry (MSGU2A) to Anonymizer having asymmetric encryption of POI and other sensitive information using Anonymizer's public key (PK_A) and a user's randomly generated key k_u.
4. On receiving the inquiry message (MSGU2A), the Anonymizer applies decryption with a private key SKA. Then, its Anonymizer cache is searched and matches I_u^2 cell identifiers for its cache contribution to get cell identifier set I_a. If the queried result found in Anonymizer's cache, it is immediately returned to the mobile user. Here, in this case, $I_u^2 = I_q - I_c - I_n$. I_u^2 represents the cell identifiers set which require to be queried in Anonymizer's cache.
5. If the queried result is not found in Anonymizer cache, then Anonymizer applies frequent pattern mining of mobile user's trajectories for users' next location prediction based on multiple constraints and performs prefetching and spatial k-anonymity according to cloaking region formation algorithm using mobile user inputs in continuous LBS. Here, in this case, $I_u^3 = I_q - I_c - I_n - I_a$. To form a cloaking region, Anonymizer needs to be queried $k - \mathrm{Num}(I_u^3)$ cells based on the cell identifier set I_u^3.
6. Finally, the Anonymizer forwards MSGA2Q consisting of cloaking region and POI to the Query Analyser using asymmetric encryption by Query Analyser PK_Q. The Query Analyser does not obtain private information (like client exact location); rather, it has the cloaking region and POI only. Query Analyser searches its database and returns the found data to Anonymizer.
7. The Query Analyzer receives the MSGA2Q, then uses a private key $SK_{Q\,to}$ for decryption and obtains the k cells in the region searches resultant POIs in these cells.
8. Anonymizer's public key PK_A, is used by Query Analyser for resultant POIs asymmetric encryption. It binds all the information in MSGQ2A and forwards it back to the Anonymizer. Private Key SK_A is used by Anonymizer to decrypts the returned message and finally, updates returned cell identifiers and POIs in its cache.
9. Anonymizer uses key k_u to perform symmetric encryption of POIs and I_u^2 to bind it in MSGA2U. Anonymizer sent back to moving client.
10. Mobile users receive MSGA2U; the mobile user decrypts this message using Key k_u and then caches this data. A cache replacement procedure is initiated whenever clients' cache becomes full for a higher cache hit ratio.

The cloaking region formulation procedure selects cells having the highest cache contribution rate. Let P_i be the query request probability by ith user in a cell and cell size is represented by m, then the relationship among these parameters can be given by contribution rate of a cache as given in the following equation.

$$\text{Cache contribution rate} = \sum_{i=1}^{\mathrm{k-Num}\left(i_u^3\right)} P_i \quad \text{where} \sum_{i=1}^{\mathrm{M \times M}} P_i = 1$$

Each cache data expires after its lifetime. After the data get expired, the system needs updating of the given cache data to improve the system cache hit ratio performance. Let t is the caching time of data in certain cells and T is the cached data average lifetime, then cell's data freshness (F) can be defined by the following equation.

$$F = 1 - \sqrt{\frac{t^2}{T^2}}, \quad \text{where T} \geq t$$

The average data freshness of the selected k – Num(I_u^3) cells can be given by the following equation.

$$\text{Average data freshness} = \frac{1}{\text{k}-\text{Num}\left(i_u^3\right)} \sum_{i=1}^{\text{k}-\text{Num}\left(i_u^3\right)} \left(1 - \sqrt{\frac{t^2}{T^2}}\right)$$

Anonymizer selects the cell (C_d) having the lowest data freshness and having the highest cache contribution rate to form the cloaking region.

$$C_d = Max \sum_{i=1}^{\text{k}-\text{Num}\left(i_u^3\right)} P_i . \frac{\sum_{i=1}^{\text{k}-\text{Num}\left(i_u^3\right)} \sqrt{\frac{t^2}{T^2}}}{\text{k}-\text{Num}\left(i_u^3\right)}$$

The formulation of a cloaking region involves the following steps.

1. According to predicted user's next location L_p, the system selects Y cells (where Y is a system parameter and Y > 2K) around L_p. Further, it chooses 2 k among Y cells having the highest query probability.
2. Among these 2 k cells, the system randomly selects W= k − Num(i_u^3) cells as a candidate set. For each candidate in W cells, the data freshness and cache contribution rate will be evaluated.
3. In the last step, the system will find cell (C_d) having the lowest data freshness and having the highest cache contribution rate to select it as a cloaking region.

5 Performance Evaluations

5.1 Simulation Setup

The client/server system modeled in our LBS is similar to that discussed in [41]. The framework for OMCPR was implemented in Java using OpenSSL 1.0.1 and experimentation work was done on a Windows 8 computer with quad-core 3.2 GHz, 64 GB RAM, and Intel i7 CPU. The public key encryption and symmetric encryptions are RSA-2048 and AES 256 Cipher Blocker Chaining respectively. The system has 10 km × 10 km as a service area which is divided into M × M cells. To deploy the coverage region in sequential pattern mining and clustering, we have used Geolife public GPS dataset collected by Microsoft Research Asia. This data set is composed of 182 users, 18,670 trajectories with approximately 1200 thousand kilometers of total distance and approx. 48,000 + hours of total duration. Every point $pt_i \in Pt_{in}$ trajectory contains pt_i.t, pt_i.lat, pt_i.lng, like timestamp, latitude, and longitude respectively. We have randomly selected 3465 different sequences for the case of the experiments. We also kept a minimum length of 10 locations and a maximum length of 15 locations that were visited from the trajectory dataset. In our experiment, the PC with specified configuration took a minimum of 18 and a maximum of 254 min for the framing of the mobility rules using sequential pattern mining with clustering. The data item access pattern follows Z_{ipf} distribution. The exponentially distributed pattern is used for the query interarrival time (d). Figure 4 depicts a part of POI scope distribution for an initial Voronoi Diagrams [37] of 10,000 distributed points. The mobile user has the facility of querying the entire neighboring user's cache lying inside 50 m. Table 4 depicts the initial value for the various used implementation parameters. The server is composed of a single process. First come first serve strategy is used in infinite queue buffer for the user's request. A negligible overhead is assumed by the system for request processing and service schedule at the server. The client's velocity uses a randomly distributed quadratic time equation and lies in between v_{min} to v_{max}. The θ is the desired threshold ($0<\theta\leq 1$) based on service quality and privacy trade-off. If θ has a higher value, then it means that the LBS compromises user privacy and satisfies the maximal query POI as requested by a mobile user.

$$v_m = 10 * \mathrm{rand}(0) + 2 * \mathrm{rand}(1) * t$$

Here, rand(0) and rand(1) lie in between 0 to 1 and are uniformly distributed. Data item size lies between S_{min} to S_{max}. The ith data item is size estimated using the following formula.

$$S_i = S_{\min} + \lfloor rand(0) * (S_{max} - S_{min}) \rfloor \quad i = 1, 2, \ldots, \text{ item\_Num};$$

5.2 Evaluation Results

We have compared proposed policy OMCPR with MobiCrowd [19], CaDSA [42] and CSKA schemes [43] on privacy and cache hit ratio parameter given in Fig. 5a, 5b. Here, we have estimated the average processing cost and communication cost [44, 45] on the initiating user query, Anonymizer, and Query Analyzer.

The system overhead is proportional to the numbers of the cell (M) and anonymity degree (k), which refers to anonymity between the special location and the user's identity. Higher is the value of k, bigger will be the cloaking region formed by Anonymizer. It provides higher data items resulting in the caches of user and Anonymizer; thus, cache hit-ratio increases. If the mobile user query counts within a time frame increase, the probability of presence in user or Anonymizer cache increases resulting in the gradual decrement in the sending of query requests to Query Analyser. If requested data found in the client's cache, then this query request needs not to send for a server which

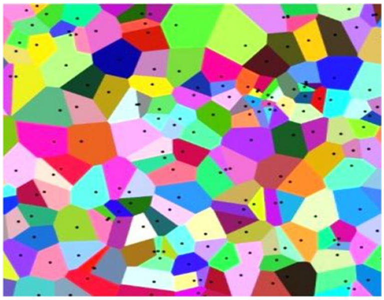
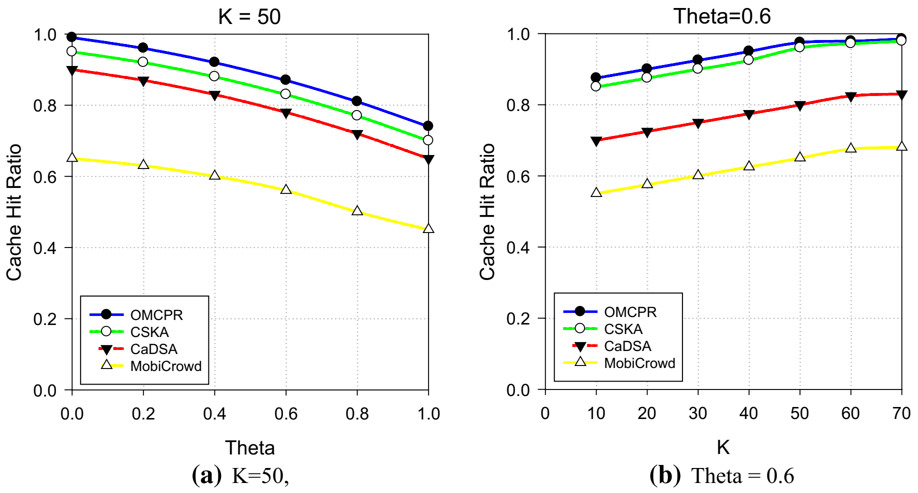


Fig. 4 Scope distribution

Table 4 Implementation model parameters' default value

Symbol	Description	Values	Symbol	Description	Values
Size_Rect	Rectangular Area size	10000 m* 10000 m	n	Number of continuous query point locations on the user's trajectory	10
Cache_SizeRatio	The ratio of the cache size to the database size	10%	Prediction_Interval	The period for evaluation of next predicted region	240.0 s
Speed_ Min	The minimum moving speed of clients	10 mps	Speed_ Max	The maximum moving speed of clients	20 mps
S_{Min}	The minimum size of the Data item	64 bytes	S_{max}	Maximum size of Data item	1024 bytes
Downlink_Band	Downlink channel bandwidth	144 kbps	Uplink_Band	Uplink channel bandwidth	4 bytes
K	Clusters count in the clustering algorithm	5	M	The number of cells	1000-10,000
Item_Num.	Number of POIs	10,000	Θ	Skewness parameter based on zipf access distribution	0.5
Interval_Query	The average time interval of queries	60.0 s	A	Most recent access item biasing weightage	0.70
R	The query range radius	0.5 km	Conf_min	Minimum confidence threshold	50%
sup_{min}	Minimum support threshold	30%	Num_ Scope	Amount of different values at various locations for each object.	220
K	Anonymity degree	10–50	O	Outlier ratio	Vary
(x1, y1)	Bottom-left vertex	(0, 0) km	(x2, y2)	Top-right vertex	(10, 10) km
N	Total count of used trajectories in the dataset	3465	Theta (θ)	The threshold value	0.1–1.0

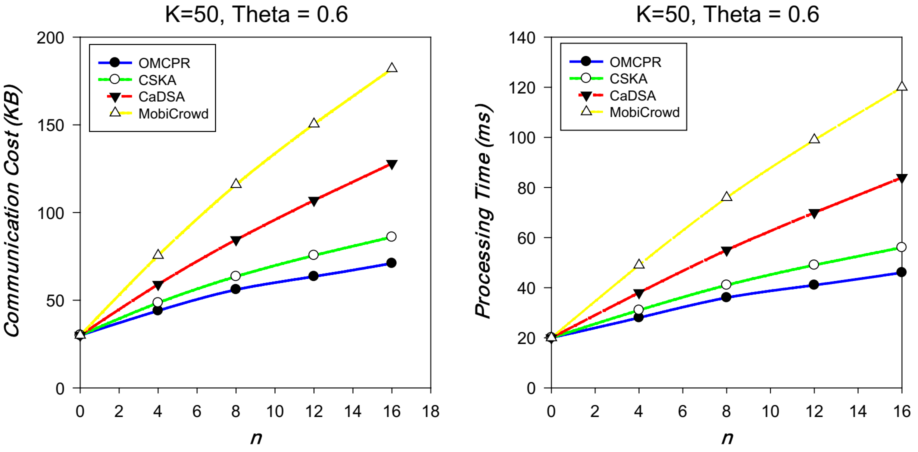


(a) K=50,

(b) Theta = 0.6

Fig. 5 Comparison of cache hit rate (user + anonymizer's cache)

leads to LBS overhead minimization and privacy improvement. Contrary to this, if most of the mobile user requests are sent to the Query Analyser, then it reduces the mobile user's privacy [46] incurring high system overhead. At M = 6000, n = 10 and k = 50, the cache hit rate of policies namely OMCPR, MobiCrowd, CaDSA, and CSKA gets down with the increase of threshold value θ. Further at M = 6000, θ = 0.6, and k = 50, the proposed OMCPR policy is better regarding communication cost and LBS application processing time overhead than that of MobiCrowd, CaDSA, and CSKA with the increase in order n. This evaluation is shown in Fig. 6a, b.



(a) Communication Cost

(b) Processing Cost

Fig. 6 Comparison of LBS server overhead

Due to incorporating cache in a trusted third party known as Anonymizer, the LBS client gets extra cache space, which leads to improvement in a cache hit ratio. The system architecture in OMCPR model follows a cooperative caching scheme for user's cache sharing, which is further a reason for a cache hit ratio improvement. The Frequent Pattern Mining of mobile user's trajectories is used for a mobile client's next location prediction. If the trajectory dataset has a high proportion of random movement i.e. higher outlier ratio then it will compromise the accuracy of users' next location prediction algorithm. The cache hit ratio of SPMC-CRP [34] and OMCPR decreases with an increase in the outlier ratio. As shown in Fig. 7a, the previous cache replacement policy PPRRP [31] outperformed SPMC-CRP after increment of a certain level of outlier ratio. Manhattan and FAR do not involve any next location prediction algorithm, so, they do not vary with the outlier ratio. Here, LBS incorporates SPMC based next location prediction algorithm which follows frequent mobility patterns mining for next location prediction. The accuracy of the next location prediction algorithm improved with increment in $\mathrm{sup}_{\mathrm{min}}$ threshold, so a cache hit ratio of SPMC-CRP and OMCPR increases with an increase in minimum support threshold $\mathrm{sup}_{\mathrm{min}}$ as shown by the graph in Fig. 7b. Mining frequent mobility patterns aim to establish the mobility rules for the current moving location of the consumer at close places. Frequent mobility rules are achieved by filtering through a confidence threshold parameter (conf_min). As the value of the confidence threshold (conf_min) increases, the lesser the number of mobility rules is formulated. The variation of a cache hit ratio with confidence threshold is depicted by the graph in Fig. 8a. So, the cache hit ratio of SPMC-CRP and OMCPR decreases with an increase in confidence threshold conf_min. As shown in Fig. 8b, increasing cache size to database size ratio improves the cache hit rato of OMCPR and previous LBS policies SPMC-CRP, PPRRP, FAR and Manhattan. From the Fig. 5a, b, 7a, b, 8a, b of graphical analysis for different LBS policy, OMCPR is better than MobiCrowd, CaDSA, MobiCrowd, SPMC-CRP, PPRRP, FAR and Manhattan in term of the cache hit ratio

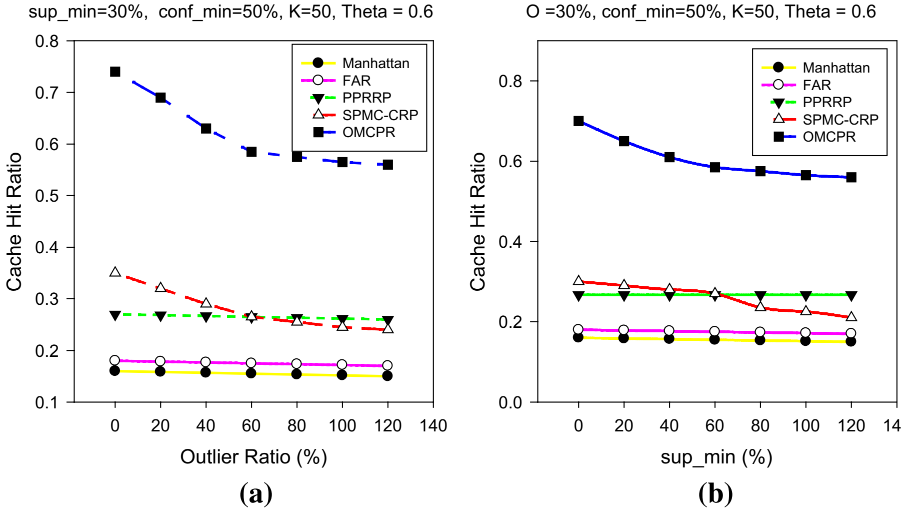


Fig. 7 (a) Cache hit ratio versus outlier ratio. **b** Cache hit ratio versus minimum support threshold ($\mathrm{sup}_{\mathrm{min}}$)

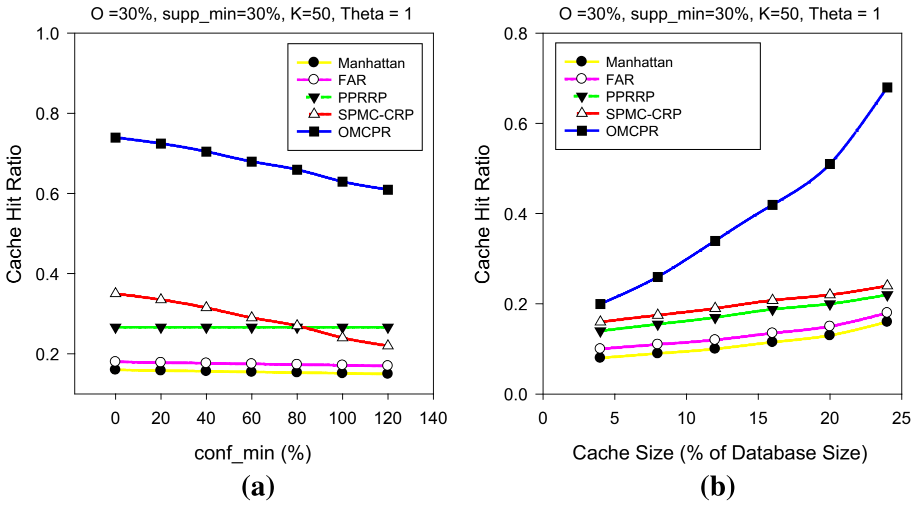


Fig. 8 (**a**) Cache hit ratio versus minimum confidence. (**b**) Threshold (conf_min) cache hit ratio versus cache size

6 Conclusion

The OMCPR model presented in this work has utilized CELP-based invalidation for the cache invalidation, SPMC-cache replacement for the cache replacement and PSKA based on mobile user's next location prediction with FMP for the mobile user's data privacy-preserving to provide high quality lossless continuous LBS. The proposed model outperformed than existing schemes by minimizing the overhead of the LBS server and higher privacy protection. Again, this model has not considered the multiple users' location prediction cases for cloaking region formation by selecting the cells. If many mobile users issue queries in LBS, the selection for the best cell to form a clocking region is a tough task and none of the previous policies has proposed any solution for the same. Therefore, the framing of the scheme supporting multiple users' expected locations to create a cloaking region can be a future research domain. Also, all the existing schemes in LBS utilizes Euclidean space for moving clients; however, in reality, the mobile users move within an underlying road network where the adversary can derive the user possible locations using knowledge of the given road network. Thus, it is necessary and important to design a user privacy preserved query processing scheme to support the anonymization of specialized locations in a given fixed road network in the next step.

Compliance with Ethical Standards

Conflict of interest The corresponding author would like to declare there is no conflict of interest situation in preparing this article.

References

1. Gupta, A. K., & Shanker, U. (2018). Location dependent information system's queries for mobile environment. In *23rd international conference on database systems for advanced applications (DASFAA) international workshops*, Gold Coast, QLD, Australia, May 21–24, pp. 1–9.
2. Ilayaraja, N., Mary Magdalene Jane, F., Safar, M., & Nadarajan, R. (2016). WARM based data prefetching and cache replacement strategies for location dependent information system in wireless environment. *Wireless Personal Communications, 90*(4), 1811–1842.
3. Xu, T., & Cai, Y. (2009). Feeling-based location privacy protection for location-based services. In *Proceedings of ACM CCS*.
4. Pingley, A., Yu, W., Zhang, N., Fu, X., & Zhao, W. (2009). Cap: A contextaware privacy protection system for location-based services. In *Proceedings of IEEE ICDCS*.
5. Manweiler, J., Scudellari, R., & Cox, L. P. (2009). Smile: Encounter-based trust for mobile social services. In *Proceedings of ACM CCS*.
6. Hu, H., & Xu, J. (2009). Non-exposure location anonymity. In *Proceedings of IEEE ICDE*.
7. Zhu, X., Chi, H., Niu, B., Zhang, W., Li, Z., & Li, H. (2013). Mobicache: When k-anonymity meets cache. In *Proceedings of IEEE GLOBECOM*.
8. Niu, B., Zhu, X., Li, W., & Li, H. (2014). Epcloak: An efficient and privacypreserving spatial cloaking scheme for LBSS. In *Proceedings of IEEE MASS*.
9. Chow, C.-Y., Mokbel, M. F., & Aref, W. G. (2009). Casper*: Query processing for location services without compromising privacy. *ACM Transactions on Database Systems, 34*(4), 1–48.
10. Zhang, S., Choo, K.-K. R., Liu, Q., & Wang, G. (2018). Enhancing privacy through uniform grid and caching in location-based services. *Future Generation Computing Systems, 86,* 881–892.
11. Niu, B., Li, Q., Zhu, X., Cao, G., & Li, H. (2014). Achieving k-anonymity in privacy-aware location-based services. In *Proceedings of the IEEE international conference on computer communications (INFOCOM '14)*, pp. 754–762, IEEE, Toronto, Canada, April–May 2014.
12. Ghinita, G., Zhao, K., Papadias, D., & Kalnis, P. (2010). A reciprocal framework for spatial K-anonymity. *Information Systems, 35*(3), 299–314.
13. Chen, J., He, K., Yuan, Q., Chen, M., Du, R., & Xiang, Y. (2018). 'Blind filtering at third parties: an efficient privacy-preserving framework for location-based services'. *IEEE Transactions on Mobile Computing, 17*(11), 2524–2535.
14. Peng, T., Liu, Q., Meng, D., & Wang, G. (2017). Collaborative trajectory privacy preserving scheme in location-based services. *Information Sciences, 387,* 165–179.
15. Liao, D., Sun, G., Li, H., Yu, H., & Chang, V. (2017). The framework and algorithm for preserving user trajectory while using location-based services in IoT-cloud systems. *Cluster Computing, 20*(3), 2283–2297.
16. Michael, K. (2004). Location-based services: A vehicle for IT & T convergence. In *Advances in e-Engineering and Digital Enterprise Technology*. Professional Engineering Publishing, University of Wollongong Research Online UK, pp. 467–477.
17. Zhao, P., Li, J., Zeng, F., Xiao, F., Wang, C., & Jiang, H. (2018). ILLIA: Enabling k-anonymity-based privacy preserving against location injection attacks in continuous LBS queries. *IEEE Internet Things Journal, 5*(2), 1033–1042.
18. Amini, S., Lindqvist, J., Hong, J., Lin, J., Toch, E., & Sadeh, N. (2011). Cache: Caching location-enhanced content to improve user privacy. In *Proceedings of ACM MobiSys*.
19. Shokri, R., Theodorakopoulos, G., Papadimitratos, P., Kazemi, E., & Hubaux, J.-P. (2014). Hiding in the mobile crowd: Location privacy through collaboration. *IEEE Transactions on Dependable and Secure Computing, 11,* 266–279.
20. Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., & Pinkerton, M. (1997). Cyberguide: A mobile context aware tour guide. *Wireless Networks, 3*(5), 421–433.
21. Takeuchi, Y., & Sugimoto, M. (2006). *CityVoyager: An outdoor recommendation system based on user location history* (pp. 625–636). Berlin: Springer.
22. Park, M.-H., Hong, J.-H., & Cho, S.-B. (2007). Location-based recommendation system using Bayesian users preference model in mobile devices. In *International conference on ubiquitous intelligence and computing*. Springer, pp. 1130–1139.
23. Horozov, T., Narasimhan, N., & Vasudevan, V. (2006). Using location for personalized poi recommendations in mobile environments. In *Proceedings of the international symposium on applications on internet*, SAINT'06, IEEE Computer Society, Washington, DC, USA, pp. 124–129.
24. Zheng, Y., Zhang, L., Xie, X., & Ma, W.-Y. (2009). Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th international conference on world wide web*, WWW'09, ACM, New York, NY, USA, pp. 791–800. https://doi.org/10.1145/1526709.1526816.

25. Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., & Ma, W.-Y. (2008). Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on advances in geographic information systems*, GIS'08, ACM, New York, NY, USA, pp. 34:1–34:10.
26. Giannotti, F., Nanni, M., Pinelli, F., & Pedreschi, D. (2007). Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, KDD'07*, ACM, New York, NY, USA, pp. 330–339.
27. Lara, O. D., Pérez, A. J., Labrador, M. A., & Posada, J. D. (2012). Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing, 8*(5), 717–729.
28. Fox, D. (2007). *Location-based activity recognition* (pp. 51–51). Berlin: Springer.
29. Wyatt, D., Philipose, M., & Choudhury, T. (2005). Unsupervised activity recognition using automatically mined common sense. In *Proceedings of the 20th national conference on artificial intelligence*—Volume 1, AAAI'05, AAAI Press, pp. 21–27
30. Zheng, V. W., Zheng, Y., Xie, X., & Yang, Q. (2010). Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th international conference on world wide web*, WWW'10, ACM, New York, NY, USA, pp. 1029–1038.
31. Kumar, A., Misra, M., & Sarje, A. K. (2008). A predicted region based cache replacement policy for location dependent data in mobile environment. *10th Inter-Research-Institute Student Semin Comput. Sci. IIIT Hyd., 7,* 1–8.
32. Kumar, A., Misra, M., & Sarje, A. K. (2007). A weighted cache replacement policy for location dependent data in mobile environments. In *SAC'07 Proceedings 2007 ACM Symposium Applied Computing*, Seoul, Repub. Korea, vol. 7, pp. 920–924.
33. Gupta, A. K., & Shanker, U. (2017). Modified predicted region based cache replacement policy for location dependent data in mobile environment. In S*ixth international conference on smart computing and comminations*, NIT Kurukshetra, India.
34. Gupta, A. K., & Shanker, U. (2017). SPMC-CRP: A cache replacement policy for location dependent data in mobile environment. In *Sixth international conference on smart computing and communications*. NIT Kurukshetra, India, pp. 632–639.
35. Zheng, B., Xu, J., Member, S., & Lee, D. L. (2002). Cache invalidation and replacement strategies for location-dependent data in mobile environments. *IEEE Transactions on Computers, 51,* 1141–1153.
36. Kumar, A., Misra, M., & Sarje, A. K. (2006). A new cost function based cache replacement policy for location dependent data in mobile environment. In *5th annual inter research institute student seminar computer science* Indian Institute Technology Kanpur, vol. 5, pp. 1–8.
37. Gupta, A. K., & Shanker, U. (2018). CELPB: A cache invalidation policy for location dependent data in mobile environment. In 22nd international database engineering applications symposium (IDEAS 2018), Calabria, Italy, June 18–20, 2018, pp. 302–306.
38. Gupta, A. K., & Shanker, U. (2020). Study of fuzzy logic and particle swarm methods in map matching algorithm. *SN Applied Sciences, 2,* 608.
39. Gupta, A. K. (2020). Spam mail filtering using data mining approach: A comparative performance analysis. In *Handling priority inversion in time-constrained distributed databases*, pp. 253–282.
40. Lubbe, C., Brodt, A., Cipriani, N., Großmann, M., & Mitschang, B. (2011). DiSCO: A distributed semantic cache overlay for location-based services. In *2011 IEEE 12th international conference on mobile data management*, Lulea, pp. 17–26.
41. Gupta, A. K., & Shanker, U. (2017). SPMC-PRRP: A predicted region based cache replacement policy. In *International conference on data and information sciences (ICDIS-2017)*, M.P, India.
42. Niu, B., Li, Q., Zhu, X., Cao, G., & Li, H. (2015). Enhancing privacy through caching in location based services. In *2015 IEEE conference on computer communications (INFOCOM)*, IEEE, pp. 1017–1025.
43. Zhang, S., Li, X., Tan, Z., Peng, T., & Wang, G. (2018). 'A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services'. *Future Generation Computing Systems, 94,* 40–50.
44. Swaroop, Vishnu, & Shanker, Udai. (2011). Concept and management issues in mobile distributed real time database. *International Journal of Recent Trends in Electrical and Electronics Engineering, 1*(1), 31–42.
45. Swaroop, V., Gupta, G. K., & Shanker, U. (2011). Issues in mobile distributed real time databases: Performance & review. *International Journal of Engineering Science and Technology (IJEST), 3*(4), 3504–3517.
46. Gupta, A. K., Prakash, S. (2018). Secure communication in cluster-based ad hoc networks: A review. In: Lobiyal, D., Mansotra, V., Singh, U. (Eds.), *Next-generation networks. Advances in intelligent systems and computing* (Vol. 638). Springer, Singapore.

Mr. Ajay K. Gupta is presently a Ph.D Research Scholar in the Department of Computer Science and Engineering of M. M. M. University of Technology, Gorakhpur 273010. His current research areas are Spatio-Temporal Database, Location Dependent Database, and Mobile Distributed Database.

Dr. Udai Shanker is presently Professor in the Department of Computer Science and Engineering of M. M. M. University of Technology, Gorakhpur-273010. For his imitation of the most modern of approaches and also for his exemplary devotion to the field of teaching, and sharing his profound knowledge with students to make better future citizen of India, he has been a role model for the new generation of academicians. Besides introduced radical and revolutionary changes that have positively impacted the database world and student community, he is a man well versed with all the intricacies of academics. He is credited with PhD from Indian Institute of Technology Roorkee and is recipient of awards from Institution of Engineers (India), Calcutta twice for his technical papers. He is authors of 80 research papers, which have been published in various National & International Journals/Conferences. He is reviewer of many International Conferences/ Journals and also Editorial Board Member of 9 International Journals. He is currently engaged in extensive research in the fields of Real Time Systems, Distributed Real Time Database Systems, Mobile Distributed Real Time Database Systems and Grid Databases.