Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Parking Recommender System Privacy Preservation through Anonymization and Differential Privacy

Yasir Saleem*[1,2] | Mubashir Husain Rehmani[3] | Noel Crespi[1] | Roberto Minerva[1]

[1]Telecom SudParis, Institut Mines-Telecom, Institut Polytechnique de Paris, 91000, Evry, France
[2]INRIA Lille−Nord Europe, 59650, Villeneuve d'Ascq, France
[3]Riomh - Intelligent Secure Systems Group, Department of Computer Science, Cork Institute of Technology, Ireland

Correspondence
*Yasir Saleem, Telecom SudParis, Institut Mines-Telecom, Institut Polytechnique de Paris, 91000, Evry, France. Email: yasir_saleem.shaikh@telecom-sudparis.eu

Summary

Recent advancements in the Internet of Things (IoT) have enabled the development of smart parking systems that use services of third-party parking recommender system to provide recommendations of personalized parking spot to users based on their past experience. However, the indiscriminate sharing of users' data with an untrusted (or semi-trusted) parking recommender system may breach the privacy because users' behavior and mobility patterns could be inferred by analyzing their past history. Therefore, in this paper, we present two solutions that preserve privacy of users in parking recommender systems while analyzing past parking history using k-anonymity (anonymization) and differential privacy (perturbation) techniques. Specifically, given an original parking database containing users' parking information, the k-anonymity mechanism constructs an anonymized database, while differential privacy perturbs the query response using the Laplace mechanism, making the users indistinguishable in both approaches, hence preserving the privacy. Experimental results on a dataset constructed from real parking measurements evaluate the trade-off between privacy and utility, therefore enabling users to receive parking spots recommendations while preserving their privacy.

KEYWORDS:
Data anonymization, differential privacy, $k-$anonymity, parking management, privacy preservation, recommender system, recommendation service, smart parking.

1 | INTRODUCTION

Recent advancements in the Internet of Things (IoT) have revolutionized our daily lives and have transformed traditional applications into smart applications. Smart parking is one example of this transition. Generally, two types of implementations are considered in smart parking systems. In the first type, the smart parking application is responsible for receiving user requests and finding the available parking spots for the user by itself. This is a widely-adopted implementation. In the second type, the smart parking application receives user requests and forwards them to the third-party recommender systems which are responsible for recommending the parking spots based on various metrics, such as traffic conditions on the roads, distance, quality of parking spots, and users' past experience [1]. The consideration of such diverse metrics into recommendations of parking spots is difficult for a smart parking application because of lack of access to diverse services and hence it is better to exploit the services of third-party recommender systems dedicated for this purpose. This implementation is currently less widely-adopted, however, is gaining attention with the horizontal and vertical emergence of IoT and smart cities applications [2], as well as with the

interoperability in IoT that interconnects various applications/deployments, hence also interconnects third-party recommender systems with the smart parking system. For instance, there is a recent EU-KR H2020 WISE-IoT project[3], that enabled the interoperability between two IoT platforms: FIWARE and oneM2M which are widely used in Europe and South Korea, respectively. It demonstrated such interoperability through a smart parking use case by adopting the second type of implementation. In this demonstration, a smart parking application operates both in Europe and South Korea. When in Europe, it connects to the FIWARE platform and recommender system to obtain the recommendations of parking spots. While when in South Korea, it connects to oneM2M infrastructure and recommender system to obtain the recommendation of parking spots[4].

In this study, our focus is on the second type of implementation. Both types of implementation consider the smart parking application to be trustworthy that is responsible for receiving user requests and maintaining a parking database. However, the second type of implementation has an additional third-party parking recommender system. Since we do not know much about the third-party parking recommender system, therefore one cannot identify its trustworthiness and it could be either trusted or semi-trusted or untrusted. The parking database contains user ID and user's current location (obtained from user's request), parking spot (obtained from recommender system), user rating (obtained from user after completing the parking) and current timestamp (the time of the user's request). The need of storing this information into the parking database is to provide personalized recommendations to users based on their past parking behavior and experience.

We focus on preserving privacy within the parking database containing users' parking history that could lead to infer users' behavior and mobility patterns. We assume that when the application sends user's current location to the parking recommender to obtain parking spot, it sends the perturbed user location by applying differential privacy (e.g., Geo-indistinguishabilty[5]), hence the parking recommender does not get the actual location of the user and the privacy is already preserved in the case of user's request. To preserve the privacy of statistical databases, there is an emerging interest in $k-$anonymity and differential privacy techniques that preserve privacy through anonymization and perturbation, respectively[6,7]. $k-$anonymity[6] is the earliest work on privacy preservation that anonymizes a dataset in such as a way that with respect to the set of quasi-identifier attributes (i.e., attributes that can identify the individuals when combined together), each record (or row) is indistinguishable from at least $k-1$ other records. Differential privacy, instead, operates on the principle of data perturbation by adding noise to the query result[7]. Therefore, the parking recommender system would not be able to differentiate among multiple records (in $k-$anonymity), as well as would not be able to find the actual query result (in differential privacy), *hence making the users unidentifiable and indistinguishable in both cases*. Both $k-$anonymity and differential privacy are formally defined and discussed in detail in Section 4.1.4 and 4.2, respectively.

Our main contribution in this paper is to preserve privacy of users in the parking recommender system while analyzing past parking history of users using k-anonymity (anonymization) and differential privacy (perturbation) techniques. Specifically, given an original parking database containing users' parking information, the k-anonymity mechanism constructs an anonymized database, while differential privacy perturbs the query response using the Laplace mechanism, making the users indistinguishable in both approaches, hence preserving the privacy. In order to evaluate the performance, we performed experiments on a dataset constructed from real parking measurements that evaluate the trade-off between privacy and utility, therefore enabling users to receive parking spots recommendations while preserving their privacy. To the best of our knowledge, these two privacy preservation techniques have not been applied and evaluated before in the perspective of preserving privacy of users in the parking database[8†].

This paper is organized as follows. Section 2 presents the related work. Section 3 presents the system and adversary models. Section 4 describes the privacy preservation techniques of k-anonymity and differential privacy. Section 5 describes the experiments for the evaluation of k-anonymity and differential privacy. Finally, section 6 concludes the paper.

2 | RELATED WORK

For privacy preservation in current smart parking systems, the major focus of existing works is on protecting real-time user's location and navigation information, cryptography, pseudonymity, encryption and consortium blockchain. The protection of historic parking database, which is the focus of our study, is not investigated much.

For instance, Ni et al.,[9,10] preserve the privacy of parking navigation using Bloom filters by enabling a user (or vehicle) to receive the navigation results, even the user is moved out of range of the queried roadside unit. They preserve the privacy using

[†]Reference[8] is the PhD thesis of YS, where some of this work is presented. See reference list for the link.

pseudonymity in which the users make queries to the cloud server which handles the parking information for available parking spots in an anonymous manner. The cloud server enables the vehicle to receive the navigation query results even if the vehicle has moved out of the range of the queried roadside unit.

Chatzigiannakis et al.,[11] preserves the privacy of a smart parking system by using public key cryptography scheme, known as elliptic curve cryptography that is suitable for resource constraint devices and is platform independent. The authors used zero knowledge proofs that avoids the exchange of confidential information, hence achieving the privacy. The authors evaluate the performance by studying the execution time and system overhead. However, the authors did not evaluate the privacy and utility of their proposed system.

Huang et al.,[12] worked on automated valet parking system for which the parking reservation is a prerequisite in order to achieve automated parking. The authors worked on preserving the private information of drivers (e.g., identity and locations) that are revealed by the reservation requests by removing the user identity and making it anonymous. However, making the users anonymous cause security problem, e.g., double-reservation attack. The authors address this security issue by allowing each anonymous user to posses only one reservation token that can be used to reserve one available parking spot. In this way, the authors claimed to preserve the privacy of user's identity and location, as well as avoid double-reservation attack using zero knowledge proofs and proxy re-signature. However, the authors mainly preserve the privacy by using pseudonymity (making the user anonymous) and did not evaluate the privacy and utility of their system.

Lu et al.,[13,14] designed a smart parking system for large parking spots using vehicular communications that offers real-time navigation and anti-theft protection. The authors preserve the privacy of users by keeping the identity of users secret, i.e., by using pseudonymity. However, only protecting the explicit identifier is not sufficient because an adversary could still identify the users uniquely by linking and disclosure attacks.

Yan et al.,[15] designed a privacy preserving parking system that relys on wireless network and sensor communications and allows users to reserve parking spots. The authors protects the privacy by using encryption technique.

Alqazzaz et al.,[16] proposed a privacy preserving and secure smart parking framework based on publish/subscribe mechanism. It provides two fold functions. Firstly, it offers the parking services, e.g., parking availability, navigation and parking reservation. Secondly, it offers security on application and network layers, as well as preserves privacy. The authors protect the privacy using encryption technique which is basically a security mechanism.

Garra et al.,[17] implemented an anonymous e-coin system that protects the privacy of a parking system and offers payment by phone without disclosing start and end time.

Hu et al.,[18] proposed a blockchain-based parking system using smart contracts that preserves privacy through a consortium blockchain in which the transactions are controlled by the legitimate nodes and are not disclosed to external entities.

Besides privacy in parking systems, many works have been done on privacy in mobility data. Although the mobility data is mostly about the trajectories and is not in the scope of parking data, but we would like to discuss some works on privacy in mobility data for the readers who are interested in considering privacy in mobility data together with parking systems. Nowadays, due to various location-based services, the users' mobility data is recorded and sensed continuously that causes a serious concern in privacy. Mobility data can disclose the users' behaviors, routines and mobility patterns. Giannotti et al.,[19] worked on mobility data and its privacy preservation by first providing the basic concepts of data privacy and then discussed the privacy in data analysis of offline mobility data. Then they presented how privacy in data mining can be achieved by design. Monreale et al.,[20] proposed a system for preserving the privacy of mobility data of trajectories using k-anonymity and generalization. Shao et al.,[21] proposed two approaches for privacy preservation using differential privacy while publishing trajectory data of ships, as well as provide comparative investigation of these two approaches. Torra[22] wrote a book on data privacy by covering the basics, developments and Big Data challenge. This book covered the perspectives of statistical and machine learning, classified privacy preservation methods, privacy risk disclosure measures and methods, masking methods and finally the information loss in privacy and utility. D'Acquisto et al.,[23] presented an overview of achieving privacy preservation by design in Big Data. This work was done in the framework of the European Union Agency for Network and Information Security (ENISA). Pratesi et al.,[24] proposed a framework for assessing the privacy risk vs. utility in data sharing systems. This system allows assessing the guarantee of data quality, as well as empirical study on privacy risk for the users in the data. Agrawal[25] highlighted the need of privacy preservation and data ownership in data mining, Hippocratic databases and sovereign information sharing.

Additionally, we have recently published a work that analyzed various Machine Learning (ML) and Deep Learning (DL) models for the prediction of availability of parking spots[26]. However, our current work is different from our previous work in a manner that our previous work is mainly about the prediction of availability of parking spots and it does not preserve the privacy of parking data. Our current paper, on the other hand, is mainly about preserving the privacy of users' parking data.

Moreover, in this paper, we apply two privacy preservation techniques: k-anonymity and differential privacy for preserving the users' privacy in parking database. However, our previously published work [26] performed a comparative analysis of four ML/DL models: Multilayer Perceptron (MLP) Neural Network, K-Nearest Neighbors (KNN), Decision Tree and Random Forest, and Ensemble Learning Approach (Voting Classifier) for the prediction of parking spot in the next thirty minutes.

The above mentioned works on privacy preservation for smart parking are firstly focused on the real-time user's location and navigation information. Secondly, they preserve privacy using pseudonymity, cryptography and encryption techniques which are prone to privacy leakage using linking and disclosing attacks, as proved in the literature. We, on the other hand, focus on privacy preservation using two well-known privacy preservation techniques of k-anonymity and differential privacy and we focus on preserving privacy within the historic parking database. k-anonymity and differential privacy are widely used in the literature. Although, they have been applied to preserve privacy in several contexts, e.g., smart grid and Internet of Things, however, we are novel in a manner that we are applying k-anonymity and differential privacy to preserve the users' privacy in parking database, and to the best of our knowledge, these privacy preservation techniques have not been applied in this context before.

3 | SYSTEM AND ADVERSARY MODELS

3.1 | System Model

Our system model is comprised of two entities: an internal smart parking system that is a trustworthy entity and an external third-party parking recommender system that is a semi-trusted or untrusted entity. The smart parking system is comprised of users, a smart parking application front-end, a service logic, a users' parking database, an anonymized database (for privacy through k-anonymity), and a perturbation mechanism (for privacy through differential privacy). The parking recommender system is a third-party recommender system that uses various metrics (such as parking and traffic information, and sensors quality) to provide recommendations. The system architecture is presented in Fig. 1. A user, registered on a smart parking application, makes a request for a nearby parking spot comprised of his user id (e.g., registration number) and current location. The smart parking application is a trusted entity that receives requests from users, forwards each user's request to the service logic entity, obtains recommended parking spots from the parking recommender and provides them to the user, as well as collects ratings from users after they have completed their parking and forward them to the service logic entity. The service logic entity is also a trusted entity and it maintains a users parking database comprised of user ids and current locations, parking spots, ratings, and timestamp attributes. The parking recommender is either an untrusted or a semi-trusted entity that receives users' current locations, analyzes their past parking history and provides parking spots recommendations. To protect the privacy of users, the parking recommender should not be able to uniquely identify the users from the users parking database. We achieve this by using two well-known privacy preservation techniques: k-anonymity and differential privacy. In k-anonymity, the service logic entity generates an anonymized version of the users parking database and releases it to the parking recommender for analysis. In differential privacy, the parking recommender makes numeric queries to the service logic entity, but instead of receiving the actual responses, the service logic entity sends the perturbed responses to the parking recommender with noise added by the Laplace mechanism.

3.2 | Adversary Model

The primary adversary[†] in our system is an untrusted (or semi-trusted) parking recommender system that needs access to the historical parking database for its recommendations of personalized and efficient parking spots. This system is susceptible to a disclosure attack, in which an adversary (i.e., the parking recommender) can recognize the behavior and mobility patterns of the users by observing the historical parking database. The adversary can track the behavior and mobility patterns of the users and uniquely identify them by analyzing the past history of users in the parking database. Such tracking could lead to the discovery of the users' private information and unique identification. For example, from past parking history, a user can be identified when he is at work, when he returns home, as well as other personal information, e.g., when and which hospitals or clinics he visits etc. Therefore, the parking database must preserve the privacy of the users such that an adversary could not be able to uniquely identify a user. We assume that the adversary is curious but not malicious.

[†]We use parking recommender system and adversary interchangeably

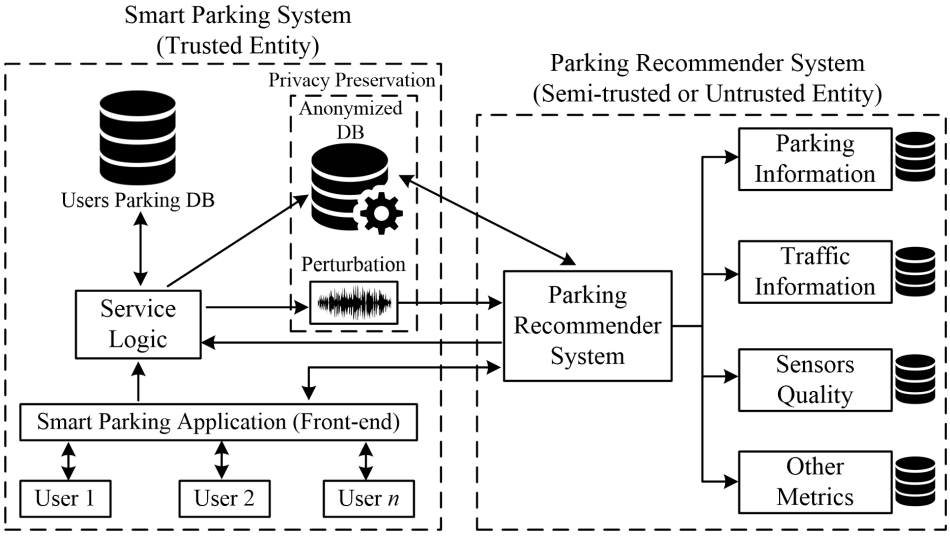


FIGURE 1 System architecture of privacy preserving parking system.

4 | PRIVACY PRESERVATION

In our considered scenario, a user, registered on a smart parking application, makes a request, comprised of user ID (e.g., registration id) and user's current location. On each user's request, the smart parking application (trusted entity) performs two fold functions. Firstly, it forwards the request to the third-party parking recommender system (semi-trusted or untrusted entity), obtains the recommended parking spot, sends it back to the user and collects the rating from the user after completing the parking. Secondly, it maintains a parking database that contains user ID and user's current location (obtained from user's request), parking spot (obtained from recommender system), user rating (obtained from user after completing the parking) and current timestamp (the time of the user's request). The sample database is presented in Table 1. This database needs to be shared with the parking recommender system for personalized recommendations of parking spots based on user's past experience. For instance, by tracking the user's parking behavior and rating, it is possible to recommend those parking spots, the users have good experience with (e.g., frequently used and highly rated). However, the parking recommender system could identify an individual and infer user's routine and mobility patterns by analyzing the user's location and parking behavior, therefore the indiscriminate sharing of user's data with parking recommender system violates the privacy of the users. The parking recommender system can easily identify a user uniquely and trace his habits, behaviours and mobility patterns by analyzing the parking database. For example, as presented in Table 1, even if we remove the user ID (the unique identifier), the recommender system could easily guess the routine of user 1, as he leaves daily in the morning from the same place (*his home*) between 8:30am to 9:00am only on weekdays (*for the work*) and parks in the similar area (*his work place*), as parking spots 3601, 3602 and 3603 are very close to each other. Hence, the parking recommender system could exploit this routine to do malicous activity, e.g., plan stealing at user's home in his absence. Here, the most important parameter for recommendation services is timestamp because the timestamp parameter helps the recommender system to learn about the user behavior and habits so that it can recommend the personalized parking spots to the users that they liked and used in the past. The timestamp is also the most critical parameter for privacy leakage because an adversary uses the timestamp for disclosure attack. For example, using the timestamp parameter, an adversary tries to find the correlation in users' parking pattern and infers the users' routines and habits. Therefore, the user's request and the database contain user private information, and sharing them in their current form with the recommender system seriously violates the privacy of users. Hence, there is a need of preserving the privacy of users. One solution is that the parking application does not share such historical database and the recommender system recommends the parkings spots only based on the real-time information. However, this will cause lack of personalized and efficient recommendations of parking spots. Another solution is that application shares the historical database by removing the user ID, however, the parking recommender system can still easily identify an individual by analyzing the other quasi-identifer attributes (e.g., user current location, parking spot and timestamp) as we discussed above. Hence, the preferred solution is to apply the privacy preservation techniques[27] so that the parking recommender system could not be able to identify the private information of the individuals.

TABLE 1 An example snapshot of data table for parking recommendation.

User ID	User Location (lon, lat)	Parking ID	User Rating	Timestamp
1	-3.80944, 43.4659	3601	5	2019-07-26 08:30:00
1	-3.80944, 43.4659	3601	5	2019-07-29 08:35:00
1	-3.80944, 43.4659	3602	5	2019-07-30 08:25:00
1	-3.80944, 43.4659	3603	4	2019-07-31 08:55:00
2	-3.80431, 43.4643	3605	1	2019-08-01 09:00:00
2	-3.79092, 43.4635	3872	5	2019-08-01 12:00:00
2	-3.80659, 43.4627	3610	3	2019-08-01 15:30:00
2	-3.79888, 43.4622	3625	4	2019-08-01 18:00:00
2	-3.79927, 43.4661	3901	4	2019-08-01 19:00:00

We preserve privacy using two techniques; one uses non-interactive data publishing through k-anonymity[6] and the other uses interactive data publishing through differential privacy[7,28]. These are discussed next in subsequent sections.

4.1 | Privacy Preservation through Anonymization

There are three widely used anonymization techniques for privacy preservation: k-anonymity, ℓ-diversity and t-closeness. The anonymization technique preserves the privacy by anonymizing the data and is applied on the microdata. The microdata is raw data that contains the information of the users, comprised of multiple attributes (or columns)[29]. The attributes in microdata are categorized into three types: i) *explicit identifiers* that can identify a user uniquely, e.g., *user id*, ii) *quasi-identifiers* that can identify a user when they are combined together, e.g., *user location, parking spot and timestamp*, iii) *sensitive attributes* are the attributes that must be protected. We do not have sensitive attributes in our system, but two examples are diseases and salary[30]. The first step in anonymization is to remove the explicit identifier.

4.1.1 | k-anonymity

$k-$anonymity[6] is the earliest work on privacy preservation that anonymizes a dataset in such as a way that with respect to the set of quasi-identifier attributes, each record (or row) is indistinguishable from at least $k-1$ other records. It achieves anonymization using generalization and suppression. The main purpose of k-anonymity is to counter against the linking attacks in which an adversary could not be able to uniquely identify a user by linking the quasi-identifier attributes (such as birthdate, zip code and gender) with external data. k-anonymity is suitable for non-interactive data publishing when there is no sensitive attribute or the distribution of sensitive attribute is sparse. In this approach, the data publisher (i.e., curator) does not want to get involve in answering all the queries and instead, releases an anonymized dataset that will be queried by the recommender systems. k-anonymity is discussed in Section 4.1.4 formally with the reasoning that why we used it instead of other anonymization techniques (i.e., ℓ-diversity and t-closeness).

4.1.2 | ℓ-diversity

k-anonymity protects from linking attacks (i.e., privacy against identifying the records), however it is susceptible to two other types of attacks of homogeneity and background knowledge attacks. In homogeneity attack, if all the sensitive attributes are same in a group of k records, the value of sensitive attribute can be identified by an adversary. In background attack, an adversary uses background knowledge to identify the individuals. To address the limitation of k-anonymity, Machanavajjhala et al.,[31] extended k-anonymity by proposing ℓ-diversity that requires each record in a group to have at least 'ℓ' diverse values for the sensitive attribute. ℓ-diversity is also suitable for non-interactive data publishing when the data publisher wants to release an anonymized dataset and does not want to get involved in answering each query. However, unlike k-anonymity, ℓ-diversity is used when the anonymized dataset should contain each record in a group to have at least 'ℓ' diverse values for the sensitive attribute. It is formally defined as:

"An equivalence group fulfills ℓ-diversity if it has at least 'ℓ' well-represented values for the sensitive attribute. A dataset having equivalence groups, all of which are ℓ-diverse, is said to be an ℓ-diverse dataset."

In brief, ℓ-diversity ensures intra-group heterogeneity of sensitive attributes by at least 'ℓ' different values. If $k{=}\ell$, ℓ-diversity automatically satisfies k-anonymity.

4.1.3 | t-closeness

Although ℓ-diversity was proposed to solve the limitations of k-anonymity, however, Li et al.,[32] proved that ℓ-diversity does not completely counter against the homogeneity attack. They used two types of attacks: skewness attack and similarity attack to demonstrate the limitation of ℓ-diversity. In skewness attack, the anonymized dataset has skewed distribution of sensitive attribute in equivalence groups and ℓ-diversity failed to prevent the attack because the distribution of the sensitive attribute is different from the dataset. In similarity attack, the anonymized dataset has distinct values of sensitive attribute in equivalence groups but they are semantically similar. ℓ-diversity also failed to prevent the attack because an adversary can estimate the value of a sensitive attribute by linking it to another sensitive attribute. These limitations of ℓ-diversity are overcome by Li et al.[32] by proposing t-closeness. t-closeness is also suitable for non-interactive data publishing and is used when a dataset that needs to be anonymized has sensitive attributes. It is suitable when the sensitive attribute has skewed distribution or distinct values in the equivalence groups of anonymized dataset. t-closeness is formally defined as:

"An equivalence group fulfills t-closeness if the distance between the distribution of a sensitive attribute in this group and that in the whole dataset is no more than a threshold t. A dataset fulfills t-closeness if all the equivalence groups have t-closeness."

4.1.4 | Privacy Preservation of Parking Data through k-anonymity

The quasi-identifier attributes in our scenario (e.g., user current location, parking spot and timestamp), while they cannot by their nature be used to uniquely identify users by linking to the external data, they can be combined together to track a user's behavior and mobility pattern (e.g., *a disclosure attack*). Therefore, we apply k-anonymity for indistinguishability among multiple users, thus preventing an adversary from identifying a user uniquely. We do not use ℓ-diversity and t-closeness because they are used when the distribution of sensitive attribute is homogeneous or skewed, respectively, however, we do not have sensitive attribute in our parking dataset and hence, k-anonymity is the most suitable candidate in our scenario. k-anonymity is formally defined as:

Definition 1 (k-anonymity). *A dataset $D(A_1, A_2, ..., A_n)$ having attributes $(A_1, A_2, ..., A_n)$, where n is the number of attributes, QI_D be the quasi-identifier attributes associated with this dataset and $D[A_1]$ is the value of A_1 attribute in dataset D. Then the dataset D satisfies k-anonymity if each sequence of values of quasi-identifier attributes in dataset D (i.e., $D[QI_D]$) appears atleast k times*[6].

The higher is the value of k, the stronger is the privacy. However, a trade-off exists between privacy and utility, the stronger is the privacy (e.g., higher value of k), the lesser will be the utility. Hence, a balance between privacy and utility is required.

k-anonymity achieves anonymization using generalization and suppression[6]. In this study, we consider *single dimensional global recoding* (i.e., mapping a value to the same level of generalization in all the records for each attribute individually). In anonymization process, removing the explicit identifiers is the first step, hence we first remove *user ID* and apply anonymization on quasi-identifier attributes of *user location (e.g., latitude and longitude)*, *parking spot* and *timestamp*, while constructing the anonymized dataset. The implementation and experimentation details are provided in Section 5.2

4.2 | Privacy Preservation through Differential Privacy

Dwork[7] coined the term differential privacy, with the definition that the outcome of a differentially private mechanism does not get highly affected by adding or removing a single record in the dataset. This mechanism can protect the privacy of users while sharing a database with an untrusted recommender system by perturbing the data. Differential privacy thus overcomes the limitations of k-anonymity, specifically the curse of dimensionality[33]. We adopt the interactive differentially private data publishing approach for numeric queries by adding noise created by the Laplace mechanism, thereby answering each numeric query f as it reaches the smart parking system (e.g., the curator) without revealing any individual record[34].

We next define differential privacy and some important notations.

Definition 2 (ϵ-Differential Privacy).

TABLE 2 Description of the experimentation database for anonymization.

Attribute	Distinct Values	Generalization type	Height
User latitude	500	25 intervals between 0.0001 and 0.05	26
User longitude	500	25 intervals between 0.0001 and 0.05	26
Timestamp	6242	Intervals of 1, 2, 3, 4, 5 and 6 hours	7
Parking spot	265	13 intervals between 10 and 300	14

A randomized process X adheres to ϵ-differential privacy if it fulfills the following two conditions: i) two adjacent datasets D_1 and D_2 differ only in one element, and ii) all outputs $S \in range(X)$ where $range(X)$ is the range of outputs of the process X [35]. Formally,

$$P_r[X(D_1) \in S] \leq e^{\epsilon} P_r[X(D_2)] \in S \tag{1}$$

where $X(D_1)$ and $X(D_2)$ are the randomized processes applied to datasets D_1 and D_2, and ϵ is a parameter of privacy, known as the privacy budget. The smaller the value of ϵ, the stronger the privacy.

Definition 3 (Sensitivity). The sensitivity defines the amount of the required perturbation. Assuming a query function $f(.)$ in a given dataset, the sensitivity $\triangle f$ is defined as:

$$\triangle f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1 \tag{2}$$

Definition 4 (Laplace Mechanism). Differential privacy uses the Laplace mechanism to perturb the results for numeric queries. It adds Laplace noise to the query result sampled from the Laplace distribution that is centered at 0 with scaling b. The Laplace noise is represented by $Lap(b)$. The higher the value of b, the higher the noise. The probability density function (pdf) of the Laplace distribution is given as $Lap(x) = \frac{1}{2b}\epsilon^{-(|x|/b)}$. The Laplace mechanism for differential privacy is formally defined as: Given a function $f : D \rightarrow \mathbb{R}$, the randomize process X adheres to ϵ-differential privacy if:

$$X(D) = f(D) + Lap(\frac{\triangle f}{\epsilon}) \tag{3}$$

Equation 3 shows that the amount of noise is dependent upon the privacy budget ϵ and sensitivity $\triangle f$. A lower privacy budget ϵ and higher sensitivity $\triangle f$ generate higher amount of noise.

The parking recommender makes the numeric query f to analyze the parking history, e.g., the rating of a selected parking spot belonging to the user's current location because it may be possible that users of a certain location did not like certain parking spots due to various reasons, e.g., too far away, crowded or a narrow or poorly-maintained road. A sample query is:

f: How many users from a specific location (user current location, e.g., 43.3905, -3.8896) gave a rating (e.g., 5-stars) for a specific parking spot (e.g., 3601) between a specific timestamp (e.g., 2019-08-01 08:00−2019-08-02 08:00)?

This type of query is used with different parameters to evaluate differential privacy in the next section.

5 | EXPERIMENTS

5.1 | Experimental Setup

We used a real parking dataset of Santander, Spain that is comprised of the occupancy time of parking spots for the month of December 2017. Real locations within Santander were then used to generate a synthetic parking dataset by assigning the user locations and ratings randomly for each record of the real parking occupancy dataset in order to evaluate the privacy preservation of k-anonymity and of differential privacy techniques. Hence, although our dataset is synthetic, it is generated from a real parking occupancy dataset and real locations, and therefore it reflects a real dataset. The size of the dataset is 15306 records composed of 500 distinct real locations as users' current locations (latitude and longitude), 265 parking spots, 6242 timestamps and ratings of between 1 to 5 for the duration of December 2017. The experiments were performed using Python 3.7.3 with NumPy v1.16.4 and Pandas v0.24.2 libraries on a macOS Catalina v.10.15 with an Intel Core i7 2.7 GHz processor with a 16 GB LPDDR3 RAM.

5.2 | Evaluation of k-anonymity

We used the four attributes (user latitude, user longitude, timestamp and parking id) presented in Table 2 of the parking dataset as Quasi-Identifier Attributes (QIA) and evaluated k-anonymity using different values of k from 2 to 750. We analyzed the performance of k-anonymity by individually studying different QIA sizes to have a complete and detailed analysis. In Section 5.2.2, we analyze QIA size = 1 by selecting *user latitude* as QIA. In Section 5.2.3, we analyze QIA size = 2 by selecting *user latitude* and *user longitude* as QIA. In Section 5.2.4, we analyze QIA size = 3 by selecting *user latitude*, *user longitude* and *parking id* as QIA. In Section 5.2.5, we analyze another case of QIA size = 3 by selecting *user latitude*, *user longitude* and *timestamp* as QIA. In Section 5.2.6, we analyze QIA size = 4 by selecting all the attributes: *user latitude*, *user longitude*, *parking id* and *timestamp* as QIA. Finally in Section 5.2.7, we present all the QIA sizes together that are discussed above to see the consolidated effect.

5.2.1 | Performance Metrics

We evaluate the performance of k-anonymity in terms of privacy and utility using six widely-adopted metrics:

1. *Average groups size* is the average size of the anonymized blocks/groups generated by the anonymization technique. It is used to measure the privacy and utility of the anonymized algorithm and has been widely used in the literature[31,32]. If the groups sizes are smaller, an adversary/analyst would be able to infer more information that enhances the utility but it weakens the privacy because due to smaller groups size, it is relatively easier to uniquely identify the users. On the other hand, when the groups sizes are larger, the privacy is stronger because it is difficult to identify the users but it reduces the utility. So, the smaller average groups size is favourable for utility while higher average groups size is favourable for privacy.

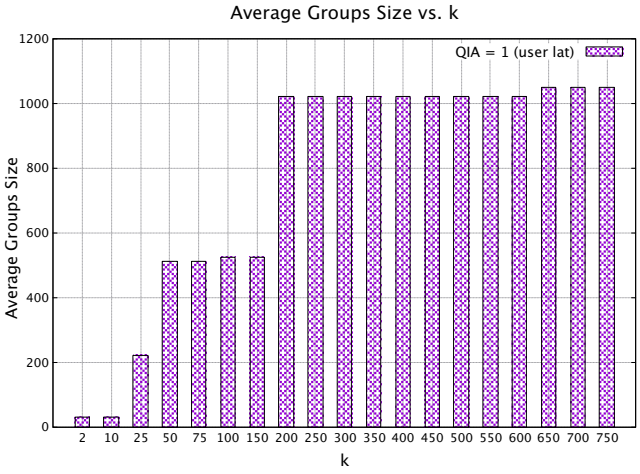
2. *Total number of groups* is the number of groups generated by the anonymization algorithm. A higher number of groups causes smaller groups size which enhances the utility but weakens the privacy. On the other hand, a lower number of groups makes the privacy stronger but it reduces the utility. So, the higher number of groups is favourable for utility while lower number of groups (or higher groups size) is favourable for privacy.

3. *Generalization height* is the height of an anonymized database, i.e., the number of generalization levels applied. It has been widely used in the literature for measuring the privacy and utility of anonymization technique[31,36,29]. With lower generalization height, the values of records are closer to their actual values, hence it enhances the utility but it weakens the privacy because an adversary/analyst could identify the users uniquely. On the other hand, when the generalization height is higher, the values of records are in the much generalized form, making it difficult for an adversary/analyst to infer useful information from the anonymized dataset which results in stronger privacy but lower utility. So, a lower generalization height is favourable for utility while higher generalization height is favourable for privacy.

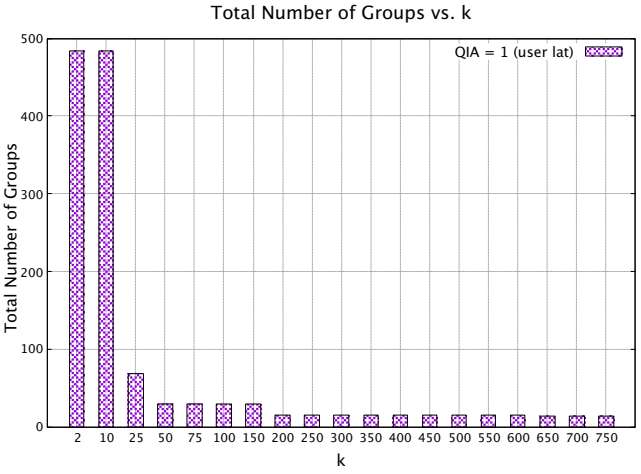
4. *Number of suppressed records* is the number of records that are suppressed because they could not fit into any anonymized block/group (because of not fulfilling the requirement of k) while privacy preservation. Suppressed records enhance privacy because if they do not get suppressed, an adversary could identify the users because of not fulfilling the requirement of k (i.e., not fulfilling the indistinguishability of records). However, they reduce the utility because the suppressed records reduce the size of the dataset, making the inference of useful information lower. So, a lower number of suppressed records is favourable for utility.

5. *Discernibility cost* is the metric to measure the indistinguishability of records with each other. The discernibility metric penalizes each record based on their indistinguishability from each other. Each unsuppressed record in a group of size j incurs a cost j, while each suppressed record incurs a cost $|D|$, i.e., the size of the original dataset D. This metric is used to measure the utility and privacy of anonymization algorithm and it has also been widely-adopted in the literature[37,31,32]. A lower discernibility cost if favourable for utility while higher discernibility cost if favourable for privacy.

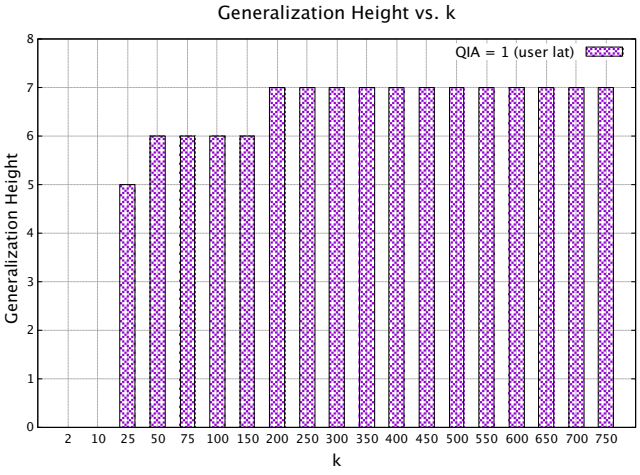
6. *Execution time* is the time required to generate an anonymized database of the original database[36]. A lower execution time is favourable.



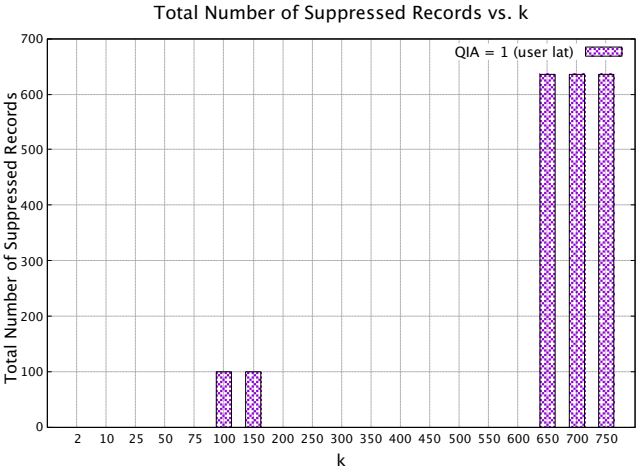
(a) Average group size



(b) Total number of groups

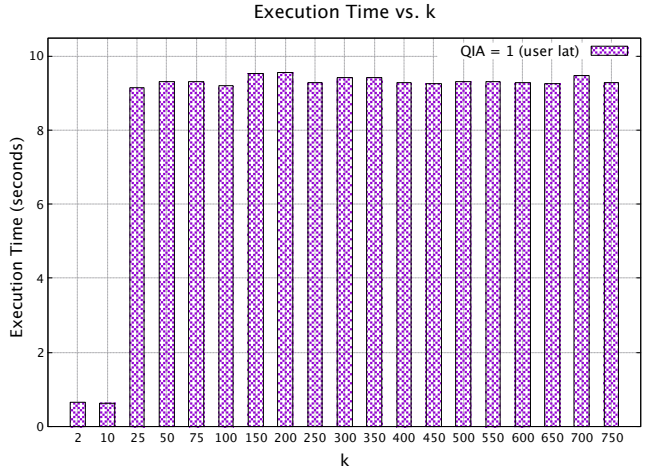


(c) Generalization height



(d) Number of suppressed records

(e) Discernibility cost



(f) Execution time

FIGURE 2 Performance evaluation of k-anonymity when QIA = 1 (user latitude)

5.2.2 | Analysis of One Quasi-Identifier Attribute

In this section, we analyze the performance of k-anonymity when one attribute is selected as QIA, i.e., QIA = 1 (user latitude).

Figure 2a presents the average groups size generated by the anonymization algorithm from k=2 to k=750. For k=2 and k=10, the average groups size is around 30. This is because no anymization is required as there is a total of 500 locations (i.e., num_loc=500) that are randomly assigned to the parking dataset D of size $|D|$=15306. Therefore, each user latitude appears around 30 times on average (i.e., $avg\_appearance = \frac{|D|}{num\_loc} \approx 30$), hence it already fulfils the requirement of k=2 and k=10, by default. When k=25, the average groups size is around 200. Although, apparently it seems that there should also be no need of anonymization of user latitude at k=25 because the expected repetition of each user latitude is 30 times (as discussed above, i.e., k=25 < $avg\_appearance$=30), however, firstly that is an average repetition appearance, and secondly, since the user locations are assigned randomly to the parking dataset, therefore the user latitude repetitions vary (i.e., from 17 to 71 times). Hence, the anonymization needs to be performed at k=25 and it generates the avarage groups size of around 200 records. From k=50 to k=150, the average groups size is around 500, and finally from k=200 to k=750, the avarage groups size is around 1000. This result shows that the average groups size increases with the increasing values of k. Higher is the value of k, higher is the group size, and hence lower is the utility.

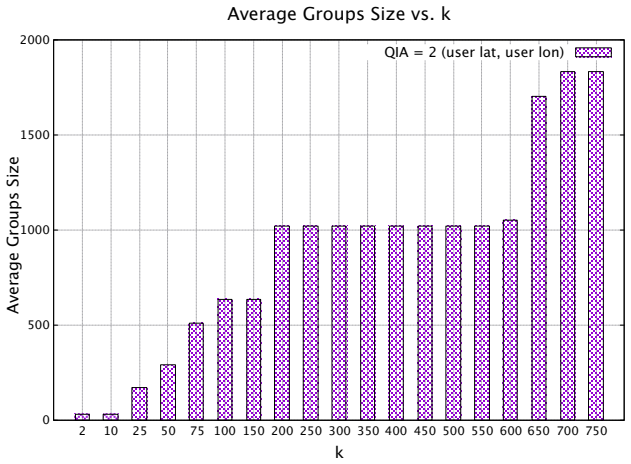
Figure 2b presents the total number of groups generated by the anonymization algorithm from k=2 to k=750. For k=2 and k=10, the total number of groups is very high and around 500. This is because no anonymization is required (as shown in Figure 2a). The total number of groups keeps decreasing from k=25 to k=750. This is because for each increasing value of k, the anonymization algorithm has to maintain indistinguishable groups of records that fulfills the requirements of k, hence causes larger groups and hence smaller number of total groups. This result shows that the total number of groups reduces with the increasing values of k. Higher is the value of k, lower is the total number of groups, and hence lower is the utility.

Figure 2c presents the generalization height applied by the anonymization algorithm from k=2 to k=750. For k=2 and k=10, the generalization height is zero because no anonymization is required (as discussed above). The generalization height for k=25 is 5 because the anonymization algorithm is able to generate anonymized parking database at this height. The generalization height for k=50 to k=150 is same and is 6 because as we analyzed in Figure 2a, since the average groups size are same from k=50 to k=150, therefore they are achieved by applying the same height of generalization. Similarly, the generalization height from k=200 to k=750 is also same and is 7. This result shows that the generalization height increases with the increasing values of k. Higher is the value of k, higher is the generalization height, and hence lower is the utility.

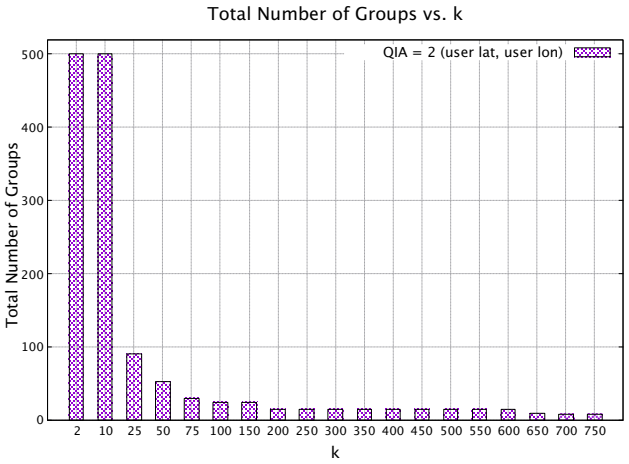
Figure 2d presents the number of suppressed records by the anonymization algorithm to generate an anonymized parking database from k=2 to k=750. The records are suppressed only when k=100,150 and when k=650,700,750. This is because in order to maximize the utility, the anonymization algorithm tries to apply as minimal generalization height as possible. Therefore, while applying a new generalization level, it first checks the number of records that are not k-anonymous (i.e., N_{non_anon}). If $N_{non\_anon} > k$, it goes for another level of generalization, otherwise if $N_{non\_anon} < k$, it suppresses these N_{non_anon} records for maximizing the utility. This is why, the number of records that are not k-anonymous (N_{non_anon}) at k=100,150 and k=650,700,750 are suppressed. This result shows that on the one hand, the number of suppressed records reduces the utility by reducing size of the dataset, but on the other hand, it actually enhances the utility by avoiding another level of generalization. Because the generalization affects the whole dataset and may reduce the utility drastically by making the records more generalized and hence more difficulty in analysis, as compared to the suppression of a small number of records (i.e., less than k).

Figure 2e presents the discernibility cost from k=2 to k=750. For k=2 and k=10, the discernibility costs are very low because no anonymization is required (as discussed in the description of Figure 2a). At k=25, the discernibility cost is around 5×10^6. An important point to note here is that from k=50 to k=150, the discernibility cost for k=100,150 is higher than that of k=50,75, while they all apply the same generalization height, have the same average group size and total number of groups, however, they differ in the number of suppressed tuples (as shown in previous Figure 2d) and this is the reason of higher discernibility cost at k=100,150 as compared to k=50,75. The similar explanation applies to the higher discernibility cost at k=650,700,750 as compared to k=200-600. This result is a very significant metric of utility and it shows that the discernibility cost increases with the increasing values of k because of higher groups size and number of suppressed records. Higher is the value of k, higher is the discernibility cost, and hence lower is the utility.

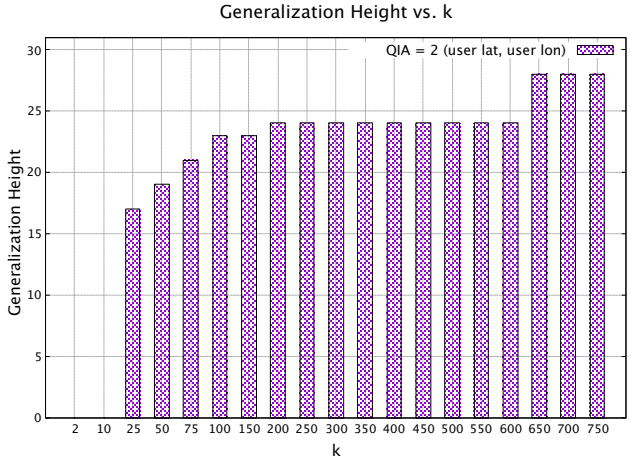
Finally, Figure 2f presents the execution time from k=2 to k=750. The execution time for k=2 and k=10 is very negligible because no anonymization is required (as discussed above). While for k=25 to k=750, the execution time is almost similar because the main execution time is consumed in making the generalizations of the records. Since, there is no much difference in the generalization heights of k=25 to k=750 (as presented in Figure 2c), therefore the execution time is similar.



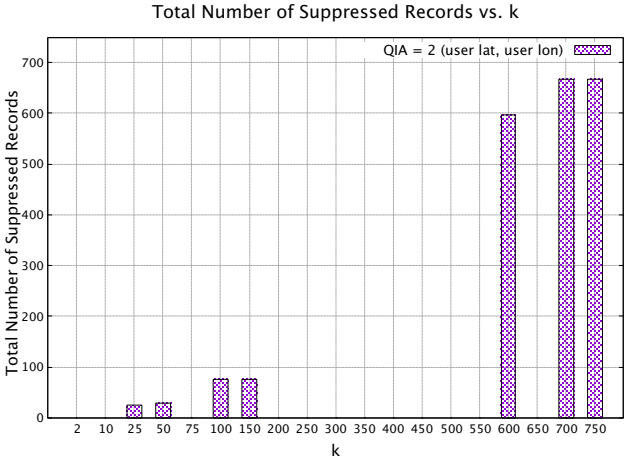
(a) Average group size



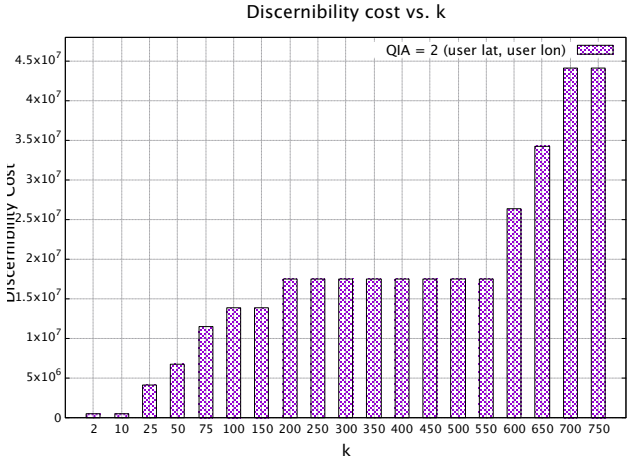
(b) Total number of groups



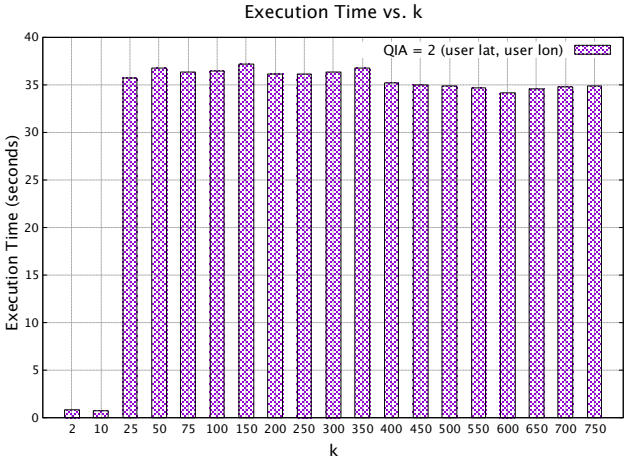
(c) Generalization height



(d) Number of suppressed records



(e) Discernibility cost



(f) Execution time

FIGURE 3 Performance evaluation of k-anonymity when QIA = 2 (user latitude, user longitude)

5.2.3 | Analysis of Two Quasi-Identifier Attributes

In this section, we analyze the performance of k-anonymity when two attribute are selected as QIA, i.e., QIA = 2 (user latitude, user longitude).

Figure 3a presents the average groups size generated by the anonymization algorithm from k=2 to k=750 when QIA size = 2 (user latitude, user longitude). For k=2 and k=10, the average groups size are around 30. Since a location is a combination of latitude and longitude, therefore the same reason discussed in the description of Figure 2a in Section 5.2.2 applies here as well, i.e., no anonymization is required because the original parking dataset already fulfills the requirement of k=2 and k=10 by default. In other words, the average appearance of each location ($avg\_appearance$=30) is greater than k=2 and k=10. When k increases from 25 to 150, the average groups size also keeps increasing to fulfill the indistinguishability requirement of k. However, from k=200 to k=600, the average groups size is similar and is around 1000. This is because at k=200, the minimim group size at k=200 is 600, therefore the higher level of anonymization (or generalization) is required above k=600. Finally, the average groups size from k=650 to k=750 are much higher and around 1700-1800. This result shows that the average groups size increases with the increasing values of k. Higher is the value of k, higher is the group size, and hence lower is the utility.

Figure 3b presents the total number of groups generated by the anonymization algorithm from k=2 to k=750 when QIA size = 2 (user latitude, user longitude). For k=2 and k=10, the total number of groups are very high and around 500. This is because no anonymization is required (as discussed before). The total number of groups keeps reducing from k=25 to k=750. This is because for each increasing value of k, the anonymization algorithm has to maintain indistinguishable groups of records that fulfills the requirements of k, hence causes larger groups and hence smaller number of total groups. This result is very similar to the result of total number of groups when QIA size = 1 (user latitude) in Figure 2b because a location is comprised of a pair of latitude and longitude. Hence, when we apply either one part of location (e.g., latitude) or both parts (e.g., latitude and longitude), we exhibit the similar trend. It shows that the total number of groups reduces with the increasing values of k. Higher is the value of k, lower is the total number of groups, and hence lower is the utility.

Figure 3c presents the generalization height applied by the anonymization algorithm from k=2 to k=750 when QIA size = 2 (user latitude, user longitude). For k=2 and k=10, the generalization height is zero because no anonymization is required (as discussed above). The generalization height for k=25 is around 17 because the anonymization algorithm is able to generate anonymized parking database at this height when anonymizing two QIA of user latitude and user longitude. The generalization height for k=50 to k=150 keeps increasing, but remains same from k=200 to k=600. The reason is similar as explain in the result of average groups size, i.e., the minimum groups size at k=200 is 600, hence no more generalization (or anonymization) is required until k=600. Finally, the generalization height from k=650 to k=750 is around 28 and is the highest. This result shows that the generalization height increases with the increasing values of k. Higher is the value of k, higher is the generalization height, and hence lower is the utility.

Figure 3d presents the number of suppressed records by the anonymization algorithm to generate an anonymized parking database from k=2 to k=750 when QIA size = 2 (user latitude, user longitude). The records are suppressed when k=25,50,100,150,600,700,750. This is because in order to maximize the utility, the anonymization algorithm tries to apply as minimal generalization height as possible. Therefore, while applying a new generalization level, it first checks the number of records that are not k-anonymous (i.e., N_{non_anon}). If $N_{non\_anon} > k$, it goes for another level of generalization, otherwise if $N_{non\_anon} < k$, it suppresses these N_{non_anon} records for maximizing the utility. This is why, the number of records that are not k-anonymous (N_{non_anon}) at k=25,50,100,150,600,700,750 are suppressed. This result shows that on the one hand, the number of suppressed records reduces the utility by reducing the size of the dataset, but on the other hand, it actually enhances the utility by avoiding another level of generalization. Because the generalization affects the whole dataset and may reduce the utility drastically by making the records more generalized and hence more difficulty in analysis, as compared to the suppression of a small number of records (i.e., less than k).

Figure 3e presents the discernibility cost from k=2 to k=750 when QIA size = 2 (user latitude, user longitude). For k=2 and k=10, the discernibility cost is very low and almost negligible because no anonymization is required (as discussed before). The discernibility cost keeps increasing from k=25 to k=150 because of having varying groups sizes. However from k=200 to k=550, the discernibility cost is same because of having the similar groups size. Although, the average group size at k=600 is also same (in Figure 3a) but it incurs higher discernibility cost because of suppressing the records. Finally, the discernibility cost increases at k=650 and stays constant from k=700 to k=750. This result shows that the discernibility cost increases with the increasing values of k because of higher groups size and number of suppressed records. When the groups size and number of suppressed records are similar, the discernibility cost is also similar (e.g., from k=200 to k=550). Higher is the value of k, higher is the discernibility cost, and hence lower is the utility.

Finally, Figure 3f presents the execution time from k=2 to k=750 when QIA size = 2 (user latitude, user longitude). The execution time for k=2 and k=10 is very negligible because no anonymization is required (as discussed above). While for k=25 to k=750, the execution time is almost similar because the main execution time is consumed in making the generalizations of the records. Since, there is no much difference in the generalization heights of k=25 to k=750 (as presented in Figure 2c), therefore the execution time is similar.

5.2.4 | Analysis of Three Quasi-Identifier Attributes (Case 1)

We have two cases when three attributes are selected as QIA. In both case, the first two QIA are *user location* and *user longitude*. In the first case, the *parking id* is selected as the third QIA, while in the second case, *timestamp* is selected as the third QIA. In this section, we consider the first case and analyze the performance of k-anonymity when three attributes are selected as QIA, i.e., QIA = 3 (user latitude, user longitude, parking id).

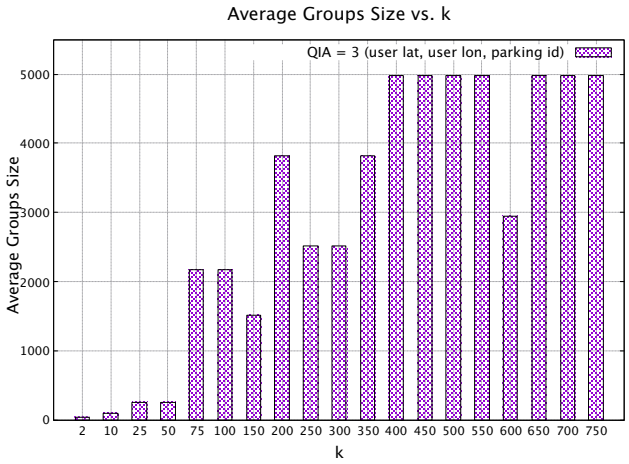
Figure 4a presents the average groups size generated by the anonymization algorithm from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, parking id). The average groups size from k=2 to k=50 are very small as compared to others because due to the repeated locations (user latitude and longitude) and parking spots, the anonymization algorithm was able to make smaller groups causing lower groups sizes. However, from k=75 to k=750, the average groups size keep increasing because since the anonymization algorithm has to fulfill the indistinguishable of records equal to k, it ended up making bigger groups. However, at k=150,250,300,600, the average groups sizes are smaller as compared to their neighbors. The reason is that at these values of k, the anonymization algorithm was able to enhance the utility by applying slightly lower generalization height (discussed next in Figure 4c) by the suppression of records (discussed next in Figure 4d). Overall, this result shows that the average groups size increases with the increasing values of k. Higher is the value of k, higher is the group size, and hence lower is the utility.

Figure 4b presents the total number of groups generated by the anonymization algorithm from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, parking id). In this result, the total number of groups are very high for k=2,10,25,50 having total number of groups around 400, 150, 60 and 60 respectively. However, at k=75, the total number of groups reduces drastically with having a total of 7 groups. From k=75 to k=750, the total number of groups are very low and between 3 to 10. The pattern is very obvious and self-explanatory. For lower values of k, the anonymization algorithm has to ensure low indistinguishability of records, and hence it can make smaller groups sizes resulting in a higher number of total groups. But as the value of k gets higher, the anonymization algorithm has to ensure high indistinguishability of records, resulting in larger groups sizes and lower number of groups. This result shows that the total number of groups reduces with the increasing values of k. Higher is the value of k, lower is the total number of groups, and hence lower is the utility.

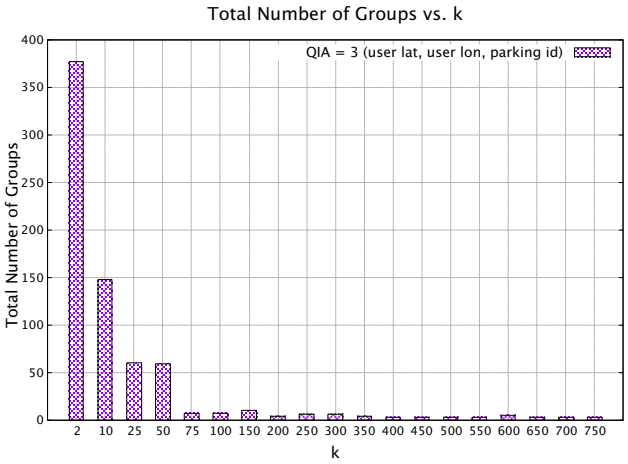
Figure 4c presents the generalization height applied by the anonymization algorithm from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, parking id). The generalization heights from k=2 to k=50 are much lower as compared to others because due to the repeated locations (user latitude and longitude) and parking spots, the anonymization algorithm was able to achieve indistinguishable records satifying the requirement of k at lower generalization heights. However, from k=75 to k=750, the generalization heights keep increasing because since the anonymization algorithm has to fulfill the indistinguishable of records equal to higher values of k, it achieved it by applying higher generalization heights. However, at k=150,250,300,600, the generalization heights are slightly lower as compared to their neighbors. This is because at these values of k, the anonymization algorithm was able to enhance the utility by applying slightly lower generalization height at the cost of suppressing the non anonymized records (N_{non_anon}) lower than k (i.e., $N_{non\_anon} < k$)(discussed next in Figure 4d). Overall, this result shows that the generalization height increases with the increasing values of k. Higher is the value of k, higher is the generalization height, and hence lower is the utility.

Figure 4d presents the number of suppressed records by the anonymization algorithm to generate an anonymized parking database from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, parking id). The records are suppressed to maximize the utility by applying as minimal generalization height as possible. The number of suppressed records are highest at k=600, i.e., around 200 more suppressed records than its neighbors, e.g., k=400-750. This is because the anonymization algorithm was able to apply one less generalization height than its neighbors, hence enhancing the utility.

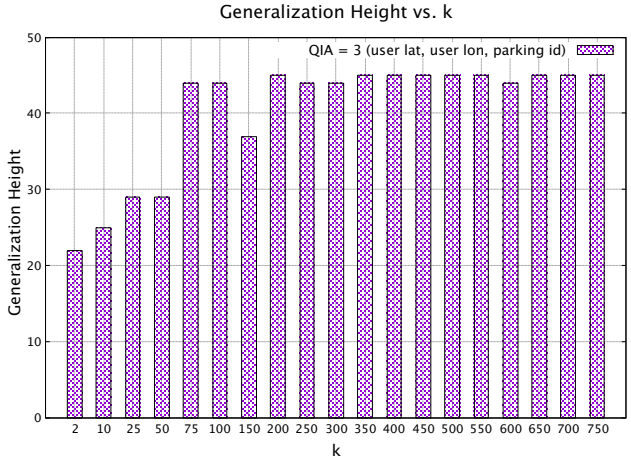
Figure 4e presents the discernibility cost from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, parking id). The discernibility costs from k=2 to k=50 are comparably quite low because the anonymization algorithm was able to make clusters of smaller groups having lower groups sizes due to the repeated locations (user latitude and longitude) and parking spots. However, from k=75 to k=750, the discernibility cost keeps increasing because since the anonymization algorithm has to fulfill the indistinguishable of records equal to k, it ended up making bigger groups and suppressing the more number of



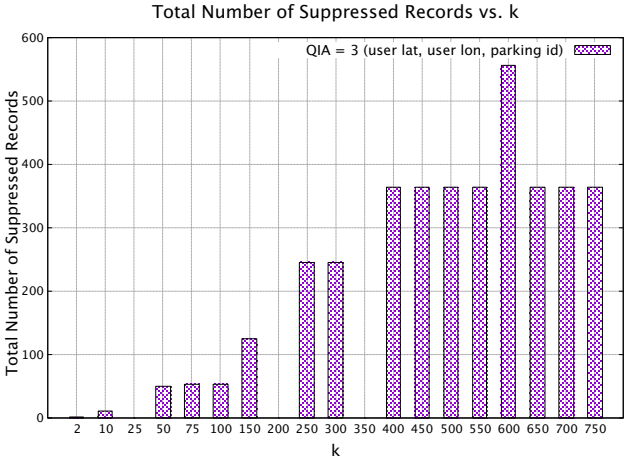
(a) Average group size



(b) Total number of groups

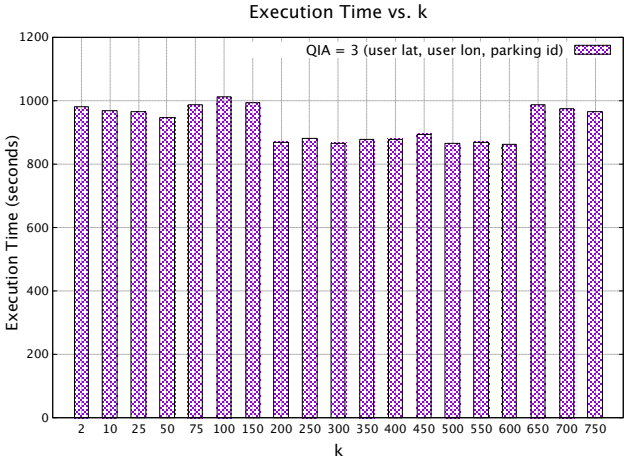


(c) Generalization height



(d) Number of suppressed records

(e) Discernibility cost



(f) Execution time

FIGURE 4 Performance evaluation of k-anonymity when QIA = 3 (user latitude, user longitude, parking id)

records. However, at k=150,250,300,600, the discernibility costs are comparable lower in the neighborhood because at these values of k, the anonymization algorithm generated smaller groups sizes by applying slightly lower generalization height. Hence, smaller groups sizes results in lower discernibility cost. Overall, this result shows that the discernibility cost increases with the increasing values of k. Higher is the value of k, higher is the discernibility cost, and hence lower is the utility.

Finally, Figure 4f presents the execution time from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, parking id). For all values of k, the execution time is almost similar because the main execution time is consumed in making the generalizations of the records. Since, there is no much difference in the generalization heights, therefore the execution time is similar.

5.2.5 | Analysis of Three Quasi-Identifier Attributes (Case 2)

In this section, we consider the second case and analyze the performance of k-anonymity when three attributes are selected as QIA, i.e., QIA = 3 (user latitude, user longitude, timestamp).

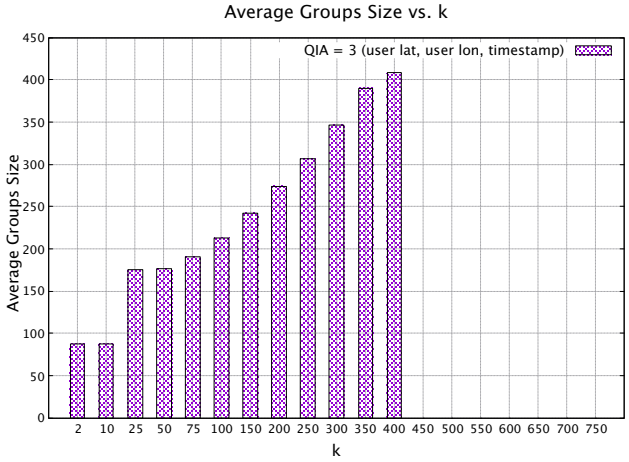
Figure 5a presents the average groups size for the second case generated by the anonymization algorithm from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, timestamp). When k=2 and k=10, the average groups size is similar and is around 90 because the anonymization algorithm was able to make smaller groups to fulfill the indistinguishability of records for lower values of k. The average groups size keeps increasing with the higher values of k because for the higher values of k, the anonymization algorithm has to maintain groups having sizes equal to or greater than the higher values of k. Another point to note here is that from k=450 to k=750, the average groups size is zero, this is because at this point (i.e., when $k >= 450$), the anonymization algorithm is unable to make an anonymized dataset from the original dataset. This result shows that the average groups size increases with the increasing values of k and the anonymization is not possible beyond $k >=$450. Higher is the value of k, higher is the group size, and hence lower is the utility.

Figure 5b presents the total number of groups generated by the anonymization algorithm from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, timestamp). In this result, the total number of groups are very high for k=2 and k=10 having a total number of groups around 160. This is because the anonymization algorithm has to maintain very smaller groups of indistinguished records, i.e., 2 and 10, hence it makes higher number of groups by creating smaller groups sizes. However, from k=25 to k=400, the total number of groups reduces drastically low because as the value of k gets higher, the anonymization algorithm has to ensure high indistinguishability of records, resulting in larger groups sizes and lower number of groups. Finally, from k=450 to k-750, the total number of groups is zero because the anonymization algorithm is unable to make an anonymized parking dataset from the original parking dataset by satisfying the requirement of k=450-700. This result shows that the total number of groups reduces with the increasing values of k. Higher is the value of k, lower is the total number of groups, and hence lower is the utility.

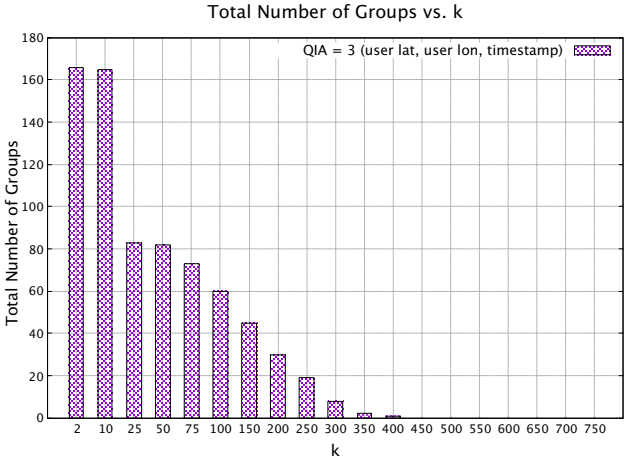
Figure 5c presents the generalization height applied by the anonymization algorithm from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, timestamp). The generalization height for k=2 to k=50 is almost similar, i.e., around 45 because the anonymization algorithm is able to make an anonymized parking dataset at this generalization height. However, from k=75 to k=450, the generalization height is around 60 and is same because this is the highest generalization height possible. At this point, the anonymization algorithm has to suppress the records (presented in next Figure 5d) because there is no more higher generalization available.

Figure 5d presents the number of suppressed records by the anonymization algorithm to generate an anonymized parking dataset from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, timestamp). The records are suppressed to maximize the utility by applying as minimal generalization height as possible. The anonymization algorithm tries to enhance the utility by applying the minimum possible generalizations and suppressing the records. The number of suppressed records from k=2 to k=50 are same because the anonymization algorithm is able to make anonymized parking dataset at the same generalization height. However, the number of suppressed records keep increasing from k=75. This is because at this point, the anonymization algorithm has applied the maximum possible generalizations (as discussed in previous figure). Hence, the only possibility is to suppress the records to achieve the k-anonymity. Here, note that from k=450 to k=750, the number of suppressed records is 15306 that is equal to the size of our original dataset. This is because the anonymization algorithm could not find an anonymization that fulfills the requirement of k=450 to k=750, hence it dropped all the records.

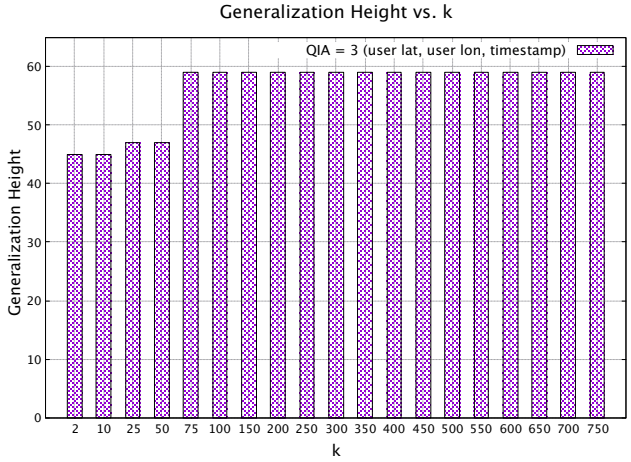
Figure 5e presents the discernibility cost from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, timestamp). The discernibility cost is another measure of utility that is dependent upon number of groups, size of groups and number of suppressed tables. For k=2 to k=50, the discernibility cost is very low and is same (results in a higher utility) because of similar number of suppressed records, and the similar ratio of number of groups to groups sizes. However, the discernibility cost keeps



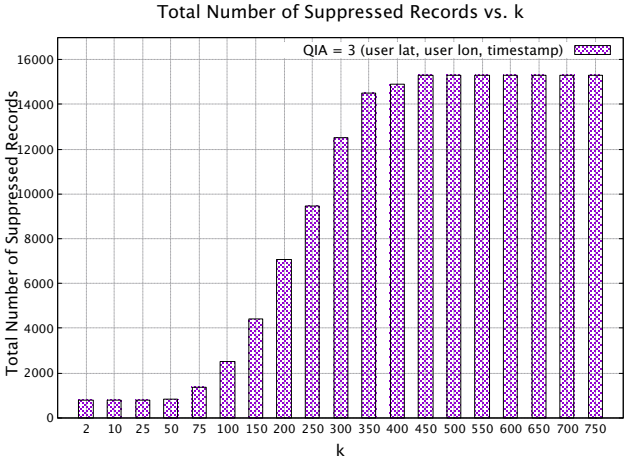
(a) Average group size



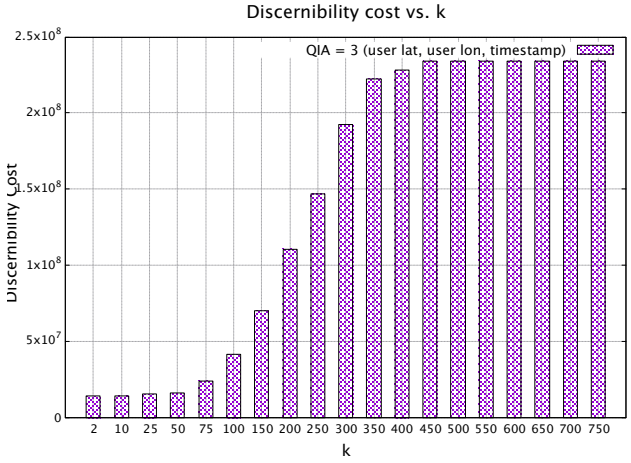
(b) Total number of groups



(c) Generalization height



(d) Number of suppressed records



(e) Discernibility cost

(f) Execution time

FIGURE 5 Performance evaluation of k-anonymity when QIA = 3 (user latitude, user longitude, timestamp)

increasing from 75 to 450 because of the phenomena described in the results of number of suppressed records, i.e., it already has applied the maximum generalizations available, hence it suppressed the records (the only possible solution) and therefore incurs higher discernibility cost (and lower utility). The discernibility cost from k=450 to k=750 is the highest possible discernibility cost (and the worst utility) because the anonymization algorithm is unable to make anonymized dataset and suppressed all the records. Overall, this result shows that the discernibility cost increases with the increasing values of k. Higher is the value of k, higher is the discernibility cost, and hence lower is the utility.

Finally, Figure 5f presents the execution time from k=2 to k=750 when QIA size = 3 (user latitude, user longitude, timestamp). For all values of k, the execution time is almost similar because the main execution time is consumed in making the generalizations of the records. Since, there is no much difference in the generalization heights, therefore the execution time is similar.

5.2.6 | Analysis of Four Quasi-Identifier Attributes

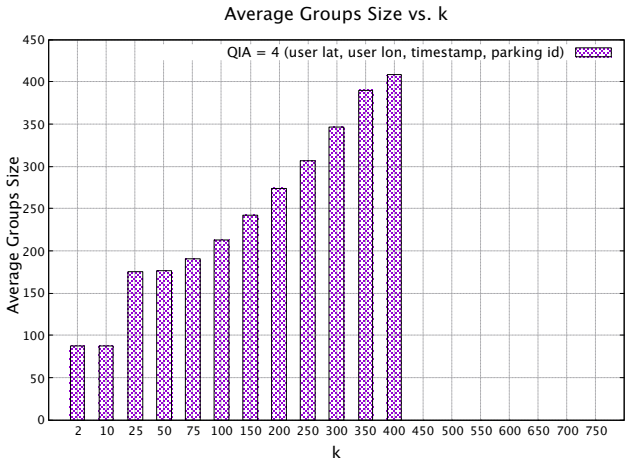
In this section, we analyze the performance of k-anonymity when all the four attributes are selected as QIA, i.e., QIA = 4 (user latitude, user longitude, timestamp and parking id).

Figure 6a presents the average groups size generated by the anonymization algorithm from k=2 to k=750 when QIA size = 4 (user latitude, user longitude, timestamp, parking id). The average groups size in this figure is very similar to the average groups size in Figure 5a (the second case of QIA=3). This is because the most heterogeneous attribute is *timestamp* having 6242 distinct values, while the *parking id* attribute is not much heterogeneous as it has 265 distinct values that is much less diverse than the *timestamp* attribute. This is why, we learned in this result that if *timestamp* and *parking id* attributes are both selected as QIA, then *parking id* attribute does not have much significance. In other words, we can say that when we select *timestamp* as QIA in anonymization, it also covers *parking id* attribute by default. To summarize the result in Figure 6a, the average groups size is very small and same at k=2 and k=10, however, it keeps increasing from k=25 to k=400 because of fulfilling the requirement of higher indistinguishability of records to satisfy higher values of k. For $k >=400$, the average groups size is zero because no anonymization exists at these points. Overall, it shows that the average groups size increases with the increasing values of k and the anonymization is not possible beyond $k >=450$. Higher is the value of k, higher is the group size, and hence lower is the utility.

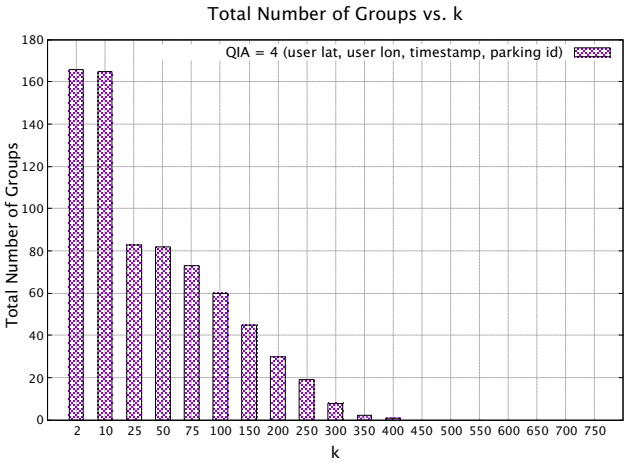
Figure 6b presents the total number of groups generated by the anonymization algorithm from k=2 to k=750 when QIA size = 4 (user latitude, user longitude, timestamp, parking id). The total number of groups in this figure is very similar to the total number of groups in Figure 5b (the second case of QIA=3). The reason is same as described above, i.e., the *timestamp* attribute is much more diverse than *parking id* attribute and hence, it already covers the *parking id* attribute in anonymization. To summarize the Figure 6b, the total number of groups are very high at k=2 and k=10 because of having lower requirement of indistinguishability of records. The total number of groups then keep reducing from k=25 to k=400 in order to fulfill the requirement of higher indistinguishability of records. From k=450 to k=750, there is no group because anonymization is not possible. Overall, the total number of groups reduces with the increasing values of k. Higher is the value of k, lower is the total number of groups, and hence lower is the utility.

Figure 6c presents the generalization height applied by the anonymization algorithm from k=2 to k=750 when QIA size = 4 (user latitude, user longitude, timestamp, parking id). The trend in this figure is similar to the trend in Figure 5c (the second case of QIA=3) but the values are different. The reason of similar trend is the same as discussed above, i.e., the *timestamp* attribute is much more diverse than *parking id* attribute and hence, it already covers the *parking id* attribute in anonymization. While, the reason of different values of generalization height is that *parking id* attribute still has to be generalized to make it indistinguishable. Otherwise, it would not be possible to generate an anonymize dataset without generalizing the *parking id* attribute. To summarize the Figure 6c, the generalization height for k=2 to k=50 is almost similar because the anonymization algorithm is able to make an anonymized parking dataset at this generalization height. However, from k=75 to k=450, the generalization height is higher and same because this is the highest generalization height possible. At this point, the anonymization algorithm has to suppress the records (presented in next Figure 6d) because there is no more higher generalization available.

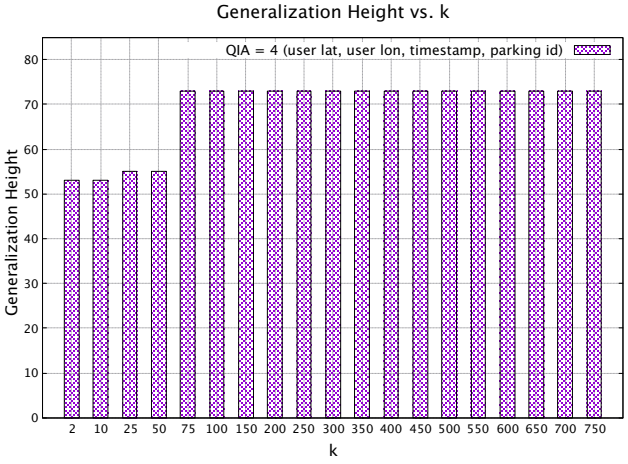
Figure 6d presents the number of suppressed records by the anonymization algorithm to generate an anonymized parking dataset from k=2 to k=750 when QIA size = 4 (user latitude, user longitude, timestamp, parking id). The total number of suppressed records in this figure is very similar to the total number of suppressed records in Figure 5d (the second case of QIA=3). The reason is similar as described above, i.e., the *timestamp* attribute is much more diverse than *parking id* attribute and hence, it already covers the *parking id* attribute in anonymization. To summarize, the number of suppressed records from k=2 to k=50 are same because the anonymization algorithm is able to make anonymized parking dataset at the same generalization height.



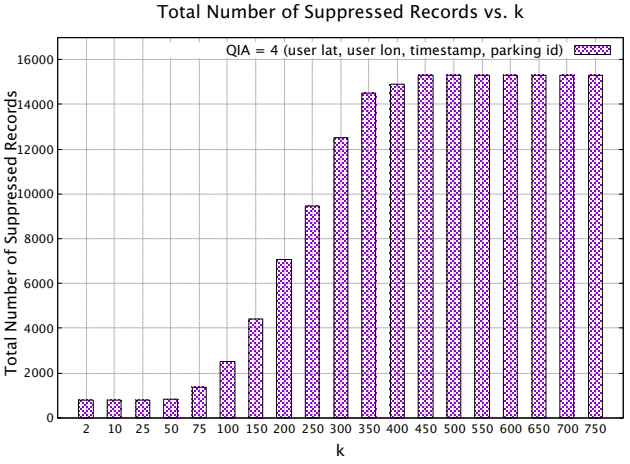
(a) Average group size



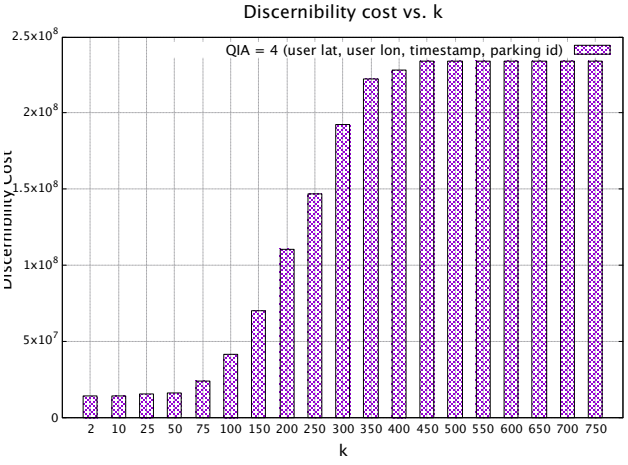
(b) Total number of groups



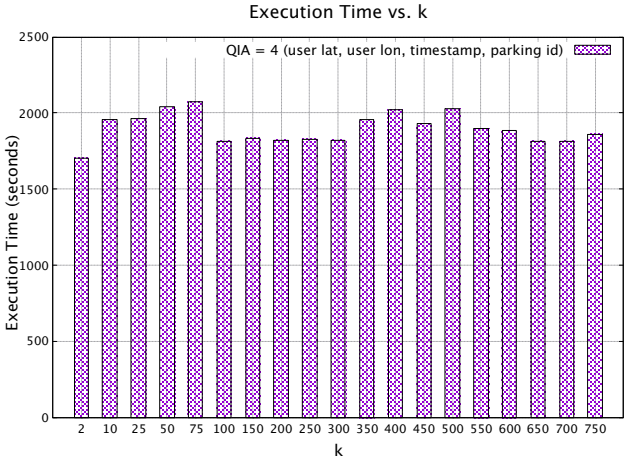
(c) Generalization height



(d) Number of suppressed records



(e) Discernibility cost



(f) Execution time

FIGURE 6 Performance evaluation of k-anonymity when QIA = 4 (user latitude, user longitude, timestamp, parking id)

However, the number of suppressed records keep increasing from k=75 because at this point, the anonymization algorithm has applied the maximum possible generalizations and follow the only possible solution of suppressing the records to achieve the k-anonymity. From k=450 to k=750, the number of suppressed records is 15306 that is equal to the size of our original dataset because the anonymization algorithm could not find an anonymization that fulfills the requirement of k=450 to k=750, hence it dropped all the records.

Figure 6e presents the discernibility cost from k=2 to k=750 when QIA size = 4 (user latitude, user longitude, timestamp, parking id). Similar to previous results, the discernibility cost in this figure is very similar to the discernibility cost in Figure 5e (the second case of QIA=3). The same explanation applies here as well. To summarize, for k=2 to k=50, the discernibility cost is very low and constant (results in a higher utility) because of similar number of suppressed records, and the similar ratio of number of groups to groups sizes. However, the discernibility cost keeps increasing from k=75 to k=450 because of the phenomena described in the results of number of suppressed records, i.e., it already has applied the maximum generalizations available, hence it suppressed the records (the only possible solution) and therefore incurs higher discernibility cost (and lower utility). Also, the discernibility cost increases with the increasing values of k. Higher is the value of k, higher is the discernibility cost, and hence lower is the utility.

Finally, Figure 6f presents the execution time from k=2 to k=750 when QIA size = 4 (user latitude, user longitude, timestamp, parking id). For all values of k, the execution time is almost similar and is around 2000 seconds because the main execution time is consumed in making the generalizations of the records. Since, there is no much difference in the generalization heights, therefore the execution time is similar.

5.2.7 | Consolidated Analysis

In this section, we present the consolidated results of all the previous analysis of k-anonymity with varying QIA sizes (i.e., QIA =1, 2, 3 (case 1 and case 2) and 4) in order to have a complete and consolidated view.

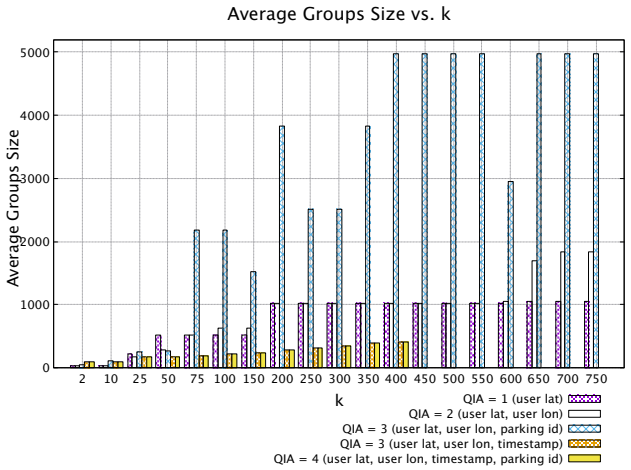
Figure 7a presents the average groups size generated by the anonymization algorithm from k=2 to k=750 for all the QIA sizes presented before. The result shows that the average groups size increases with the higher values of k. However, there is a surprising behaviour of the first case of QIA = 3 (user latitude, user longitude, parking id). It makes much average higher groups sizes as compared to other QIA sizes (i.e., QIA = 1, 2, 3 (case 2) and 4). This is because of non suppression of records, i.e., QIA = 3 (case 1) does not suppress the records and hence, it results in larger groups sizes, while QIA = 3 (case 2) and QIA = 4 suppress the records, causing smaller groups sizes. The main insight is that the groups sizes increases with the increasing values of k, as well as with the increasing QIA sizes at a limit when no suppression is made. Overall, the average groups size increases with the increasing values of k. Higher is the value of k, higher is the groups size, and hence lower is the utility.

Figure 7b presents the total numbers of groups generated by the anonymization algorithm from k=2 to k=750 for all the QIA sizes presented before. There are two insights gained from this result. Firstly, the total number of groups reduces with the increasing values of k. Secondly, the total number of groups also reduces with the increasing QIA sizes. Higher is the value of k and QIA sizes, lower is the number of groups, and hence lower is the utility.

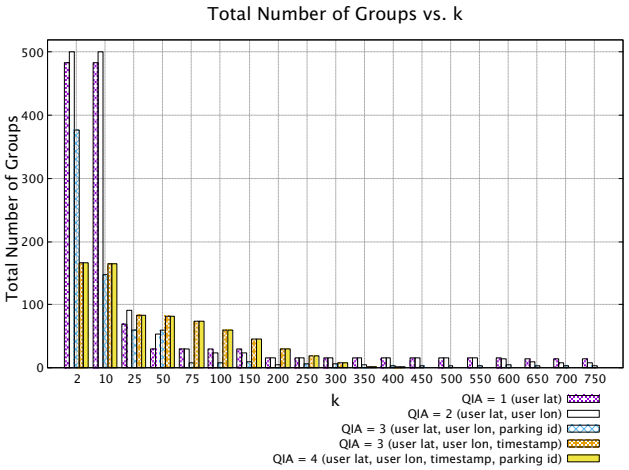
Figure 7c presents the generalization heights applied by the anonymization algorithm from k=2 to k=750 for all the QIA sizes presented before. There are three insights gained from this result. Firstly, the generalization height increases with the higher values of k. Secondly, the generalization height also increases with the higher QIA sizes. Thirdly, the generalization heights increases until k=75. After k=75, the generalization height is same because no more higher generalization is available. Higher is the value of k and QIA sizes, higher is the generalization height, and hence lower is the utility.

Figure 7d presents the number of suppressed records by the anonymization algorithm from k=2 to k=750 for all the QIA sizes presented before. There are four insights gained from this result. Firstly, the number of suppressed records increases with the higher values of k. Secondly, the number of suppressed records also increases with the higher QIA sizes. Thirdly, the number of suppressed records for QIA = 3 (case 2) and QIA = 4 increases until k=400. From k >=450, the number of suppressed records is equal to the size of the original dataset, i.e., no anonymization is made. Fourthly, the number of suppressed records by QIA = 3 (case 2) and QIA = 4 is same, it means that they exhibit the same behaviour. Higher is the value of k and QIA sizes, higher is the number of suppressed records, and hence lower is the utility.

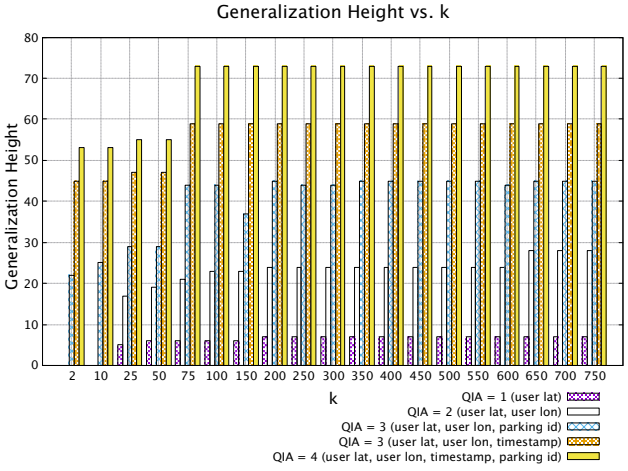
Figure 7e presents the number of discernibility cost incurred by the anonymization algorithm from k=2 to k=750 for all the QIA sizes presented before. There are four insights gained from this result. Firstly, the discernibility cost increases with the higher values of k. Secondly, the discernibility cost also increases with the higher QIA sizes. Thirdly, the discernibility cost for QIA = 3 (case 2) and QIA = 4 increases until k=400. From k >=450, the discernibility is equal and is the highest possible discernibility cost because all the records in the datatset are suppressed. Fourthly, the number of suppressed records by QIA = 3



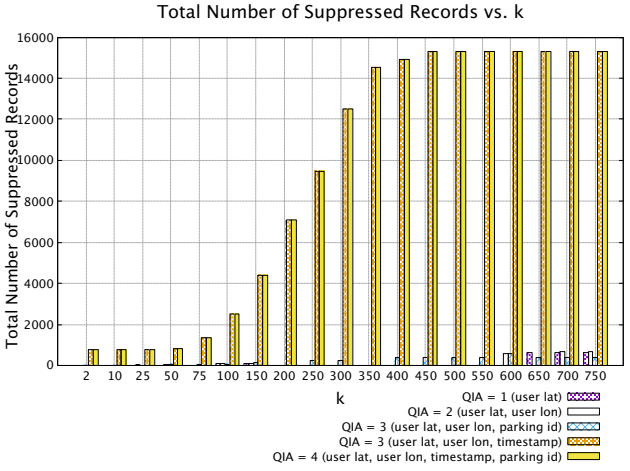
(a) Average group size



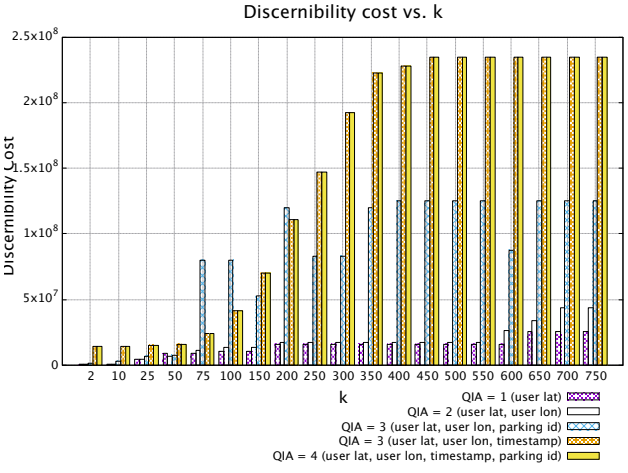
(b) Total number of groups



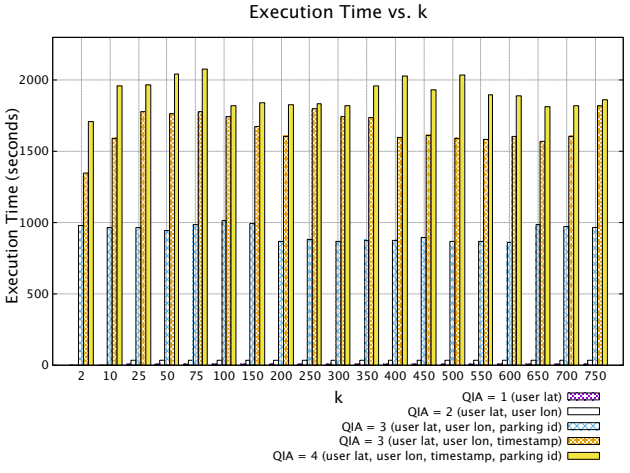
(c) Generalization height



(d) Number of suppressed records



(e) Discernibility cost



(f) Execution time

FIGURE 7 Performance evaluation of k-anonymity for all QIA = 1, 2, 3, 4

(case 2) and QIA = 4 is same and they exhibit the same behaviour. Higher is the value of k and QIA sizes, higher is the number of discernibility cost, and hence lower is the utility.

Finally, figure 7f presents the execution time by the anonymization algorithm from k=2 to k=750 for all the QIA sizes presented before. There are two insights gained from this result. Firstly, the execution time increases with the higher values of k. Secondly, for each QIA size, the execution time is almost similar. It means that the execution time is mainly dependent upon the size of QIA, or in other words, it depends upon the level of generalizations that need to be applied to construct an anonymized parking dataset.

5.3 | Evaluation of Differential Privacy

We evaluated differential privacy for the numeric query type as discussed in Section 4.2 by generating 1000 random queries. For each query, the user' current location, timestamp, parking spot and rating are randomly selected from the parking dataset. Subsequently, a time range is randomly selected between 1 to 30 days and for each location, we consider nearby locations within 5km radius. We evaluate differential privacy using different values of privacy budget ϵ from ϵ=0.1 to ϵ=1.0. The sensitivity $\triangle f$ defines the number of records that gets affected with the addition or removal of a user. As in our parking dataset, a user may appear multiple times but we do not know the exact number of appearences, therefore, we analyze the effects of different values of sensitivity $\triangle f$ values from 1 to 5, i.e., the appearence or removal of a user in the dataset affects 1 to 5 records, respectively.

5.3.1 | Performance Metrics

We evaluate the accuracy and privacy of differential privacy by using two widely-adopted metrics:

- *Mean Absolute Error (MAE)* measures the average amount of errors. It is an average over the number of queries of the absolute differences between the actual query result and noisy result. It measures the privacy and the utility. If the MAE is high, the difference between the actual and noisy query results is high that makes the privacy stronger but reduces the utility. While if the MAE is low, the difference between the actual and noisy query results is low that enhances the utility but weakens the privacy. MAE has been widely-adopted in the literature for evaluating differential privacy[38,39,40]. It is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|r_{a,i} - r_{n,i}| \tag{4}$$

  where N is the total number of queries, $r_{a,i}$ is the actual response of query i, $r_{n,i}$ is the noisy response of query i.

- *Root Mean Square Error (RMSE)* is a quadratic scoring function and it also measures the average amount of errors. It is the square root of the average of squared differences between the actual query result and noisy result. It measures the privacy and the utility. Similar to MAE, if the RMSE is high, the difference between the actual and noisy query results is high that makes the privacy stronger but reduces the utility. While if the RMSE is low, the difference between the actual and noisy query results is low that enhances the utility but weakens the privacy. It has been widely-adopted in the literature for evaluating differential privacy[38,39,40,41]. It is defined as:

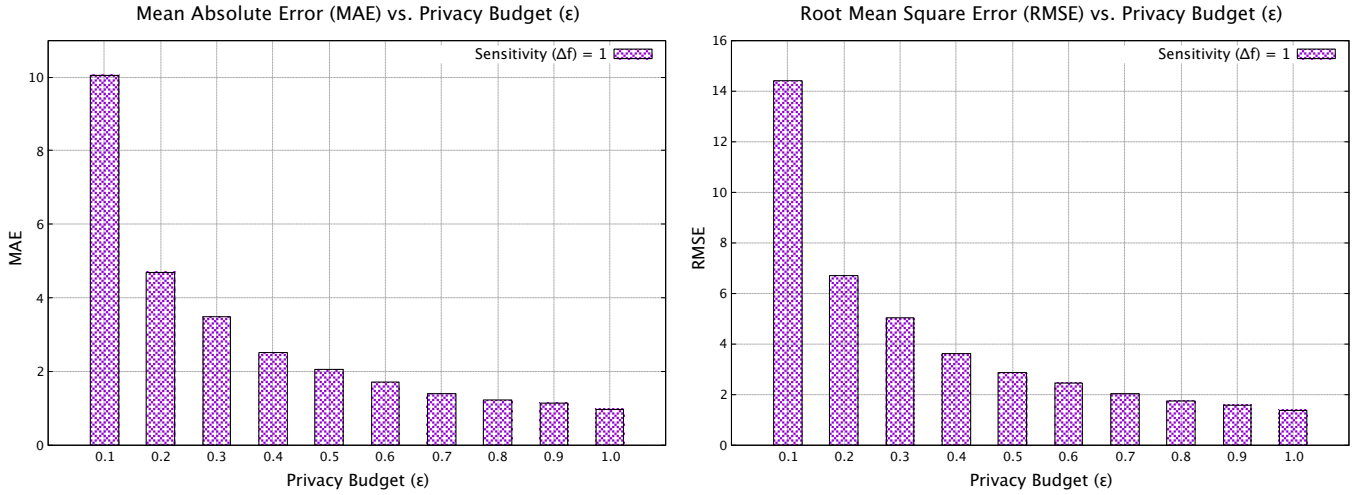
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(r_{a,i} - r_{n,i})^2}{N}} \tag{5}$$

  where N is the total number of queries, $r_{a,i}$ is the actual response of query i, $r_{n,i}$ is the noisy response of query i.

5.3.2 | Analysis of Individual Sensitivities

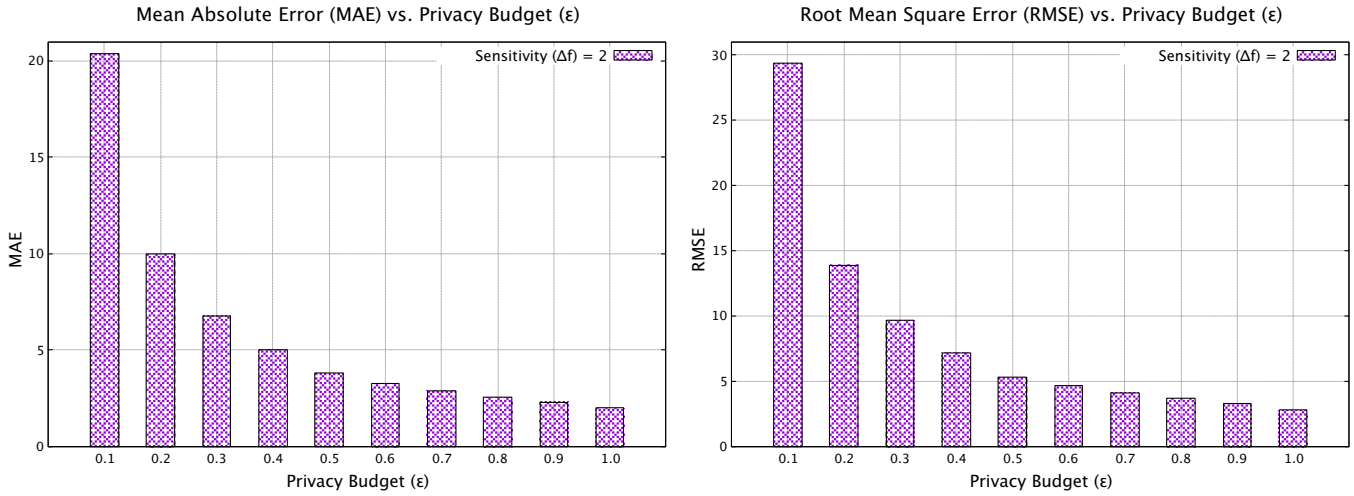
In this section, we analyze the performance of differential privacy in terms of accuracy and privacy using MAE and RMSE for privacy budget ϵ=0.1 to ϵ=1.0 by analyzing each sensitivity ($\triangle f$) individually.

Figure 8 presents MAE and RMSE for privacy budget ϵ=0.1 to ϵ=1.0 when sensitivity $\triangle f$=1 (i.e., the addition or removal of a user affects one record in the parking dataset). It shows that when ϵ=0.1, the MAE and RMSE are very high, i.e., 10 and 14, respectively because ϵ=0.1 guarantees the highest privacy, however at the cost of the worst utility. However, as ϵ keeps



(a) MAE **(b)** RMSE

FIGURE 8 MAE and RMSE for varying privacy budget ϵ when sensitivity ($\triangle f$)=1



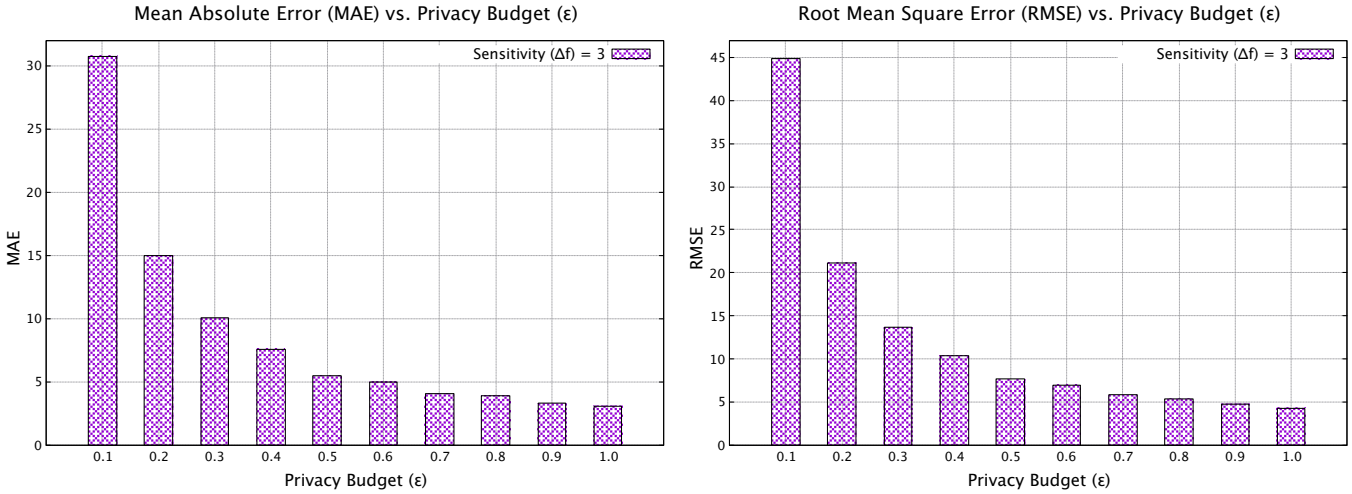
(a) MAE **(b)** RMSE

FIGURE 9 MAE and RMSE for varying privacy budget ϵ when sensitivity ($\triangle f$)=2

increasing, the MAE and RMSE keeps reducing drastically and at ϵ=1.0, both MAE and RMSE are close to zero. It means that we have the highest utility at this point, however there is no privacy because the noisy results are very similar to the actual results. Overall, the MAE and RMSE reduces with the increasing values of privacy budget ϵ. Higher is the privacy budget ϵ, higher is the privacy but lower is the utility. Inversely, lower is the privacy budget ϵ, lower is the privacy but higher is the utility.

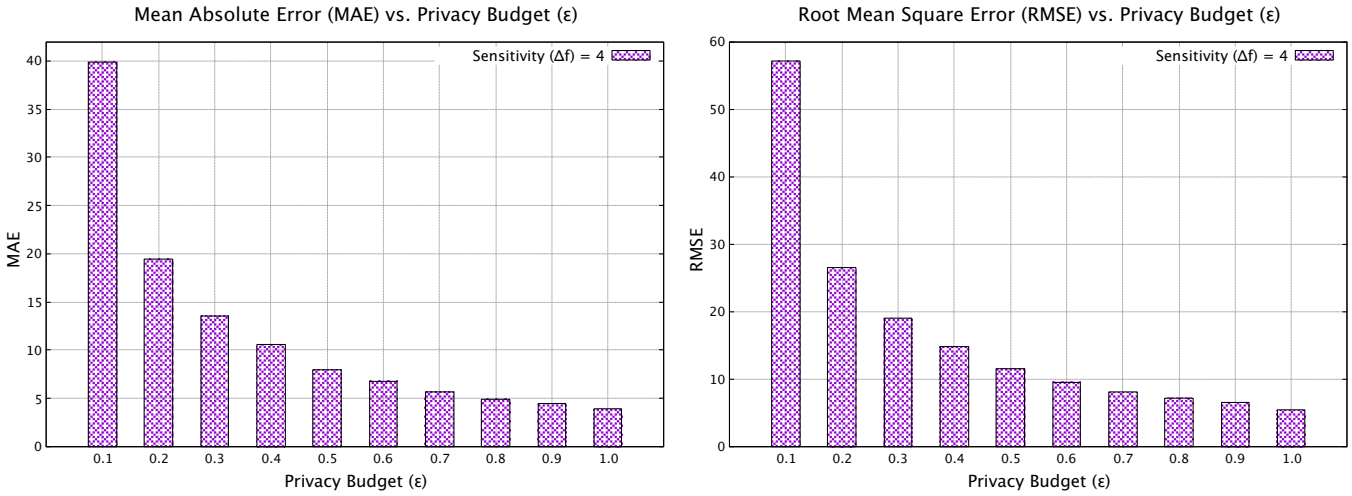
Figure 9 presents MAE and RMSE for privacy budget ϵ=0.1 to ϵ=1.0 when sensitivity $\triangle f$=2 (i.e., the addition or removal of a user affects two records in the parking dataset). The trend is very similar to Figure 8 when $\triangle f$=1, however, at ϵ=0.1 the MAE and RMSE are almost double. This is because when $\triangle f$=2, since an additional or removal of a user affects at least two records, therefore we have two times (i.e., 2×) MAE and RMSE than at $\triangle f$=1. However, the MAE and RMSE at $\triangle f$=2 keeps reducing with the increasing values of privacy budget ϵ and at ϵ=1.0, the MAE and RMSE are almost similar to those at $\triangle f$=1.

Similarly, figures 10, 11 and 12 present MAE and RMSE for privacy budget ϵ=0.1 to ϵ=1.0 when sensitivity $\triangle f$=3, $\triangle f$=4 and $\triangle f$=5 (i.e., the addition or removal of a user affects three, four and five records in the parking dataset), respectively. The trends are similar as discussed before. Initially, at ϵ=0.1, the MAE and RMSE are three, four and five times (3×, 4× and 5×)



(a) MAE **(b)** RMSE

FIGURE 10 MAE and RMSE for varying privacy budget ϵ when sensitivity ($\triangle f$)=3



(a) MAE **(b)** RMSE

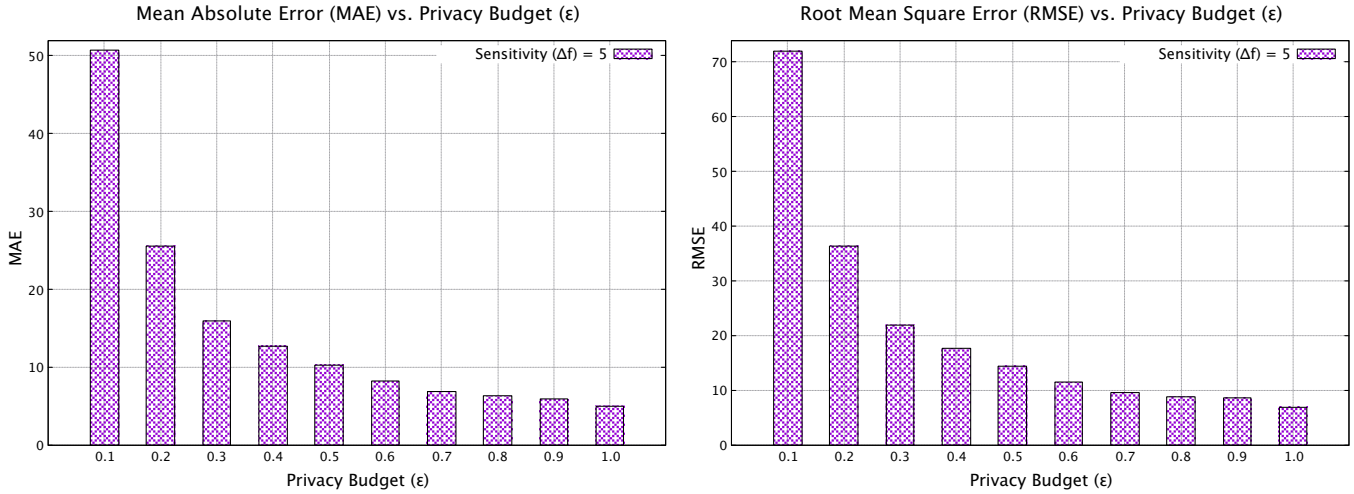
FIGURE 11 MAE and RMSE for varying privacy budget ϵ when sensitivity ($\triangle f$)=4

for $\triangle f$=3, 4, 5, respectively, as compared to MAE and RMSE for $\triangle f$=1. However, as ϵ increases and reaches towards 1.0, the MAE and RMSE get close to zero. Overall, the MAE and RMSE are very high at privacy budget ϵ=0.1, which give very strong privacy, however at the cost of the worst utility. The MAE and RMSE keep reducing with the increasing values of privacy budget ϵ and at ϵ=1.0, the MAE and RMSE are very similar for all sensitivities $\triangle f$=3, 4, 5.

5.3.3 | Consolidated Results

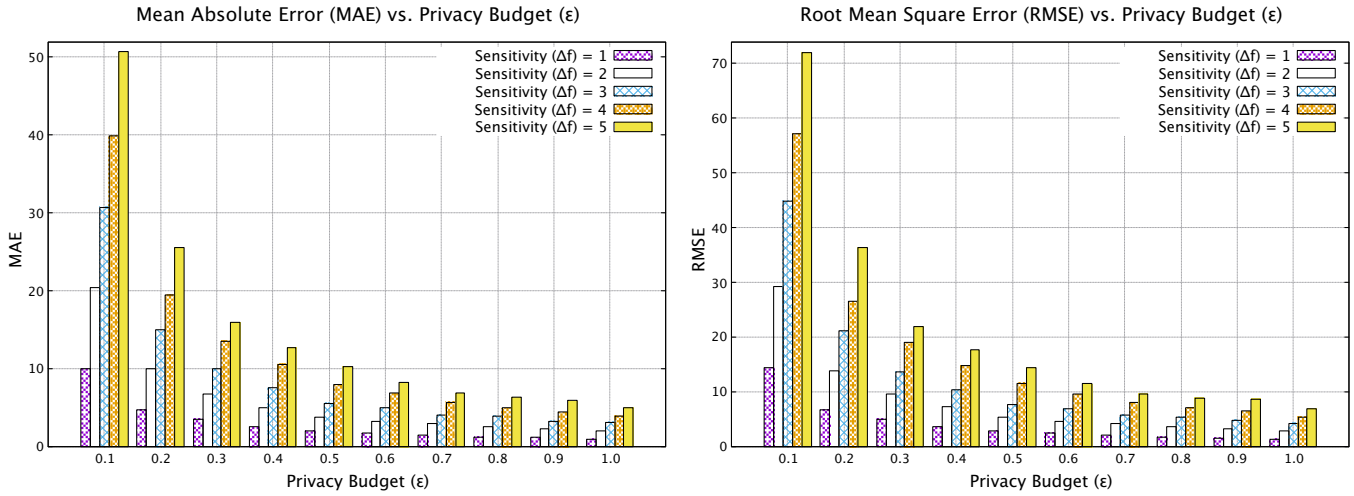
In this section, we present the consolidated results of all the previous analysis of differential privacy with all previously discussed sensitivity $\triangle f$ values (i.e., $\triangle f$=1, 2, 3, 4, 5) in order to have a complete and consolidated view.

Figure 13 presents the consolidated results of MAE and RMSE for all sensitivity values $\triangle f$=1, 2, 3, 4, 5 for privacy budget ϵ=0.1 to ϵ=1.0. These results provide two insights. Firstly, initially at ϵ=0.1, the MAE and RMSE are very high for all sensitivity $\triangle f$ values (that provides very strong privacy but no utility), however, as privacy budget ϵ keeps getting closer to 1.0, the MAE



(a) MAE **(b)** RMSE

FIGURE 12 MAE and RMSE for varying privacy budget ϵ when sensitivity ($\triangle f$)=5



(a) MAE **(b)** RMSE

FIGURE 13 MAE and RMSE for varying privacy budget ϵ when sensitivity ($\triangle f$)=1, 2, 3, 4, 5

and RMSE are becoming similar and getting close to zero (that provides very high utility but no privacy). Secondly, as the sensitivity $\triangle f$ value increases, the MAE and RMSE also gets higher to many folds but then they get closer to other sensitivity $\triangle f$ values when privacy budget ϵ values gets higher. Overall, the MAE and RMSE reduces with the increasing values of privacy budget ϵ. Higher is the privacy budget ϵ, higher is the privacy but lower is the utility. Inversely, lower is the privacy budget ϵ, lower is the privacy but higher is the utility.

5.3.4 | Summary

This section summarizes the findings of the experiments on privacy preservation through k-anonymity and differential privacy. We found that k-anonymity is suitable for smaller values of k and for lower QIA sizes. When k is much higher, the utility is very low. Specifically, when k >=450 and QIA = 3 (user latitude, user longitude, timestamp) or QIA size = 4 (user latitude, user longitude, timestamp, parking id), the k-anonymity is unable to generate an anonymized parking dataset because the requirement

of k does not get fulfilled. Also, the behavior of QIA = 3 (user latitude, user longitude, timestamp) and QIA = 4 is very similar because *timestamp* attribute is much more diverse than *parking id* attribute and therefore, it covers *parking id* in anonymization by default. For differential privacy, we found that when the privacy budget ϵ is very low (e.g., ϵ=0.1), the privacy is very strong, however the utility is worse. As the privacy budget ϵ keeps getting higher, the utility starts improving, however at the cost of weakening the privacy. Additionally, the sensitivity $\triangle f$ also affects the privacy and utility. The higher is the sensitivity $\triangle f$ value, stronger is the privacy but lower is the utility. Moreover, we considered one parking dataset in our experiments which is presented in Table 1. However, the experiments can be reproduced with other datasets because the other parking dataset will also have the similar attributes, as our considered attributes are those attributes that are required in almost all the smart parking systems. Therefore, we believe that there will not be any problem in reproducing the evaluation with other parking dataset.

6 | CONCLUSION

In this paper, we preserve the privacy of users while sharing their historical parking information (which contains their private behavior and mobility patterns) with a semi-trusted or untrusted third-party parking recommender system through two well-known privacy preservation techniques of anonymization and perturbation: k-anonymity and differential privacy. The proposed implementations preserve privacy of users while receiving parking spots recommendations based on their past parking experience. We discuss the system and adversary models, discussion and applicability of k-anonymity and differential privacy on parking dataset. Extensive experimental results evaluated the impact of utility and privacy of both privacy preservation techniques on our parking dataset.

For future works, since we preserved the privacy of users against parking recommender by anonymizing and perturbing the parking database. Therefore, when we go for privacy, we lose the data of individual users and correlation between the records, hence making it no longer possible to provide personalized recommendation with respect to the individual user's habits and preferences. This ultimately affects the quality of recommendations. Hence, there is a need to study the impact of privacy preservation on recommendations services. Additionally, Blockchain also provides security and privacy through through its inherent features, such as applying strong cryptographic algorithms and hashing techniques. Hence, another future work is to explore the use of Blockchain as a tool for security and privacy in smart parking system. Furthermore, it would be interesting to evaluate using other datasets and see their impact.

Conflict of interest

The authors declare that there is no conflict of interests relevant to this article.

References

1. Sedjelmaci SAH, Brahmi IH, Ansari N, Rehmani MH. Cyber Security Framework for Vehicular Network based on a Hierarchical Game. *IEEE Transactions on Emerging Topics in Computing* 2019; In Press: 1–1. doi: 10.1109/tetc.2018.2890476

2. Saleem Y, Crespi N. Mapping of Sensor and Route Coordinates for Smart Cities. *IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* 2018: 570–576. doi: 10.1109/COMPSAC.2018.00087

3. Worldwide Interoperability for Semantic IoT (WISE-IoT), Last accessed: December 2019, http://wise-iot.eu/en/home/. .

4. Sotres P, Torre CLDL, Sanchez L, Jeong S, Kim J. Smart City Services over a Global Interoperable Internet-of-Things System: The Smart Parking Case. *Global Internet of Things Summit (GIoTS)* 2018(March). doi: 10.1109/GIOTS.2018.8534546

5. Andrés ME, Bordenabe NE, Chatzikokolakis K, Palamidessi C. Geo-indistinguishability: Differential Privacy for Location-based Systems. *Proceedings of the ACM Conference on Computer and Communications Security* 2013: 901–914. doi: 10.1145/2508859.2516735

6. Sweeney L. k-anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Puzziness and Knowledge-Based Systems* 2002; 10(5): 557–570.

7. Dwork C. Differential Privacy. In: Bugliesi M, Preneel B, Sassone V, Wegener I. , eds. *Automata, Languages and Programming* 2006 (pp. 1–12).

8. Shaikh YS. *Privacy Preserving Internet of Things Recommender Systems for Smart Cities*. Theses. Institut Polytechnique de Paris, 2020. https://tel.archives-ouvertes.fr/tel-02500640.

9. Ni J, Zhang K, Yu Y, Lin X, Shen X. Privacy-Preserving Smart Parking Navigation Supporting Efficient Driving Guidance Retrieval. *IEEE Transactions on Vehicular Technology* 2018; 67(7): 6504–6517. doi: 10.1109/TVT.2018.2805759

10. Ni J, Zhang K, Lin X, Yu Y, Shen XS. Cloud-Based Privacy-Preserving Parking Navigation Through Vehicular Communications. *International Conference on Security and Privacy in Communication Systems* 2016: 85–103. doi: 10.1007/978-3-319-59608-2

11. Chatzigiannakis I, Vitaletti A, Pyrgelis A. A Privacy-preserving Smart Parking System using an IoT Elliptic Curve based Security Platform. *Computer Communications* 2016; 89-90: 165–177. doi: 10.1016/j.comcom.2016.03.014

12. Huang C, Lu R, Lin X, Shen X. Secure Automated Valet Parking: A Privacy-Preserving Reservation Scheme for Autonomous Vehicles. *IEEE Transactions on Vehicular Technology* 2018; 67(11): 11169–11180. doi: 10.1109/TVT.2018.2870167

13. Lu R, Lin X, Zhu H, Shen X. SPARK: A New VANET-Based Smart Parking Scheme for Large Parking Lots. *IEEE INFOCOM* 2009: 1413–1421. doi: 10.1109/INFCOM.2009.5062057

14. Lu R, Lin X, Zhu H, Shen X. An Intelligent Secure and Privacy-Preserving Parking Scheme Through Vehicular Communications. *IEEE Transactions on Vehicular Technology* 2010; 59(6): 2772–2785. doi: 10.1109/TVT.2010.2049390

15. Yan G, Yang W, Rawat DB, Olariu S. SmartParking: A Secure and Intelligent Parking System. *IEEE Intelligent Transportation Systems Magazine* 2011; 3(1): 18–30.

16. Alqazzaz A, Alrashdi I, Aloufi E, Zohdy M, Ming H. SecSPS: A Secure and Privacy-Preserving Framework for Smart Parking Systems. *Journal of Information Security* 2018; 9(4): 299–314. doi: 10.4236/jis.2018.94020

17. Garra R, Mart S. A Privacy-Preserving Pay-by-Phone Parking System. *IEEE Transactions on Vehicular Technology* 2017; 66(7): 5697–5706.

18. Hu J, He D, Zhao Q, Choo KKR. Parking Management: A Blockchain-Based Privacy-Preserving System. *IEEE Consumer Electronics Magazine* 2019; 8(4): 45–49. doi: 10.1109/MCE.2019.2905490

19. Giannotti F, Monreale A, Pedreschi D. Mobility Data and Privacy. In: Renso C, Spaccapietra S, Zimanyi E., eds. *Mobility Data: Modeling, Management, and Understanding*Cambridge University Press. 2013 (pp. 174–193).

20. Monreale A, Andrienko GL, Andrienko NV, et al. Movement Data Anonymity through Generalization. *Transactions on Data Privacy* 2010; 3(2): 91–121.

21. Shao D, Jiang K, Kister T, Bressan S, Tan KL. Publishing Trajectory with Differential Privacy: A Priori vs. A Posteriori Sampling Mechanisms. In: Springer. ; 2013: 357–365.

22. Torra V. *Data Privacy: Foundations, New Developments and the Big Data Challenge*. Springer . 2017.

23. D'Acquisto G, Domingo-Ferrer J, Kikiras P, Torra V, Montjoye dYA, Bourka A. Privacy by Design in Big Data: An Overview of Privacy Enhancing Technologies in the Era of Big Data Analytics. *arXiv preprint arXiv:1512.06000* 2015.

24. Pratesi F, Monreale A, Trasarti R, Giannotti F, Pedreschi D, Yanagihara T. PRUDEnce: A System for Assessing Privacy Risk vs Utility in Data Sharing Ecosystems. *Transactions on Data Privacy* 2018; 1: 139-167.

25. Agrawal R. Data Privacy. In: Boulicaut JF, Esposito F, Giannotti D., eds. *Machine Learning: ECML 2004*Oxford: Springer. 2004 (pp. 266-290).

26. Awan FM, Saleem Y, Minerva R, Crespi N. A Comparative Analysis of Machine/Deep Learning Models for Parking Space Availability Prediction. *Sensors* 2020; 20(1): 322.

27. Wang T, Zheng Z, Rehmani MH, Yao S, Huo Z. Privacy Preservation in Big Data from the Communication Perspective-A Survey. *IEEE Communications Surveys and Tutorials* 2019; 21(1): 753–778. doi: 10.1109/COMST.2018.2865107

28. Hassan MU, Rehmani MH, Chen J. Differential Privacy Techniques for Cyber Physical Systems: A Survey. *IEEE Communications Surveys and Tutorials* 2019: 1–44.

29. Samarati P. Protecting Respondents' Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering* 2001; 13(6): 1010–1027.

30. Li T, Li N, Zhang J, Molloy I. Slicing: A New Approach for Privacy Preserving Data Publishing. *IEEE Transactions on Knowledge and Data Engineering* 2012; 24(3): 561–574. doi: 10.1109/TKDE.2010.236

31. Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M. l-Diversity: Privacy Beyond k-Anonymity. *22nd International Conference on Data Engineering (ICDE)* 2006: 24–35.

32. Li N, Li T, Venkatasubramanian S. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. *IEEE 23rd International Conference on Data Engineering* 2007(3): 106–115.

33. Salas J. Sanitizing and Measuring Privacy of Large Sparse Datasets for Recommender Systems. *Journal of Ambient Intelligence and Humanized Computing* 2019: 1–12. doi: 10.1007/s12652-019-01391-2

34. Zhu T, Li G, Zhou W, Yu PS. Differentially Private Data Publishing and Analysis: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 2017; 29(8): 1619–1638. doi: 10.1109/TKDE.2017.2697856

35. Soria-Comas J, Domingo-Ferrer J, Sanchez D, Megias D. Individual Differential Privacy: A Utility-Preserving Formulation of Differential Privacy Guarantees. *IEEE Transactions on Information Forensics and Security* 2017; 12(6): 1418–1429. doi: 10.1109/TIFS.2017.2663337

36. LeFevre K, DeWitt DJ, Ramakrishnan R. Incognito: Efficient full-domain K-anonymity. *ACM SIGMOD International Conference on Management of Data* 2005: 49–60. doi: 10.1145/1066157.1066164

37. Bayardo RJ, Agrawal R. Data Privacy Through Optimal k-Anonymization. *21st International Conference on Data Engineering (ICDE)* 2005: 217—-228.

38. Fan W, He J, Guo M, Li P, Han Z, Wang R. Privacy Preserving Classification on Local Differential Privacy in Data Centers. *Journal of Parallel and Distributed Computing* 2020; 135: 70–82. doi: 10.1016/j.jpdc.2019.09.009

39. Su X, Sperli G, Moscato V, Picariello A, Esposito C, Choi C. An Edge Intelligence Empowered Recommender System Enabling Cultural Heritage Applications. *IEEE Transactions on Industrial Informatics* 2019; 15(7): 4266–4275. doi: 10.1109/TII.2019.2908056

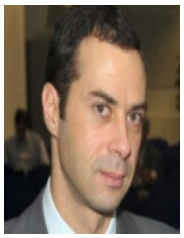
40. Bobadilla J, Ortega F, Hernando A, Gutiérrez A. Recommender Systems Survey. *Knowledge-Based Systems* 2013; 46: 109–132. doi: 10.1016/j.knosys.2013.03.012

41. Yu J, Gao M, Rong W, Song Y, Xiong Q. A Social Recommender Based on Factorization and Distance Metric Learning. *IEEE Access* 2017; 5: 21557–21566. doi: 10.1109/ACCESS.2017.2762459

AUTHOR BIOGRAPHY

Yasir Saleem received the B.S. degree in Information Technology from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2012, and the M.Sc. degrees in Computer Science by Research from Sunway University, Malaysia, and Lancaster University, U.K. (under a dual degree program) in 2015. He is currently pursuing the Ph.D. degree with the Service Architecture Laboratory, Institut Mines-Telecom, Telecom SudParis, France. His research interests include Internet of Things, Semantic Web, social Internet of Things, cognitive radio networks, and wireless sensor networks. He served in the TPC for IEEE MELECON 2016 and FMEC 2016 conferences He is also a Reviewer of various journals, such as the IEEE Wireless Communications, the IEEE Communications Magazine, Pervasive and Mobile Computing, Ad Hoc Networks, Computer Networks, the Journal of Network and Computer Applications, Artificial Intelligence Review, the IEEE ACCESS, Wireless Networks, and many others. He also has been a Reviewer for various conferences, such as IEEE ICC 2013, IEEE Globecom 2014, IWCMC 2015, MICC 2015, ICIN 2017, and ISNCC 2017. http://www.yasirsaleem.com/

Mubashir Husain Rehmani (M'14-SM'15) received the B.Eng. degree in computer systems engineering from Mehran University of Engineering and Technology, Jamshoro, Pakistan, in 2004, the M.S. degree from the University of Paris XI, Paris, France, in 2008, and the Ph.D. degree from the University Pierre and Marie Curie, Paris, in 2011. He is currently working as Assistant Lecturer at in the Department of Computer Science, Cork Institute of Technology, Ireland. Prior to this, he worked as Post Doctoral Researcher at the Telecommunications Software and Systems Group (TSSG), Waterford Institute of Technology (WIT), Waterford, Ireland. He also served for five years as an Assistant Professor at COMSATS Institute of Information Technology, Wah Cantt., Pakistan. He is currently an Area Editor of the IEEE Communications Surveys and Tutorials. He served for three years (from 2015 to 2017) as an Associate Editor of the IEEE Communications Surveys and Tutorials. Currently, he serves as Associate Editor of IEEE Communications Magazine, Elsevier Journal of Network and Computer Applications (JNCA), and the Journal of Communications and Networks (JCN). He is also serving as a Guest Editor of Elsevier Ad Hoc Networks journal, Elsevier Future Generation Computer Systems journal, the IEEE Transactions on Industrial Informatics, and Elsevier Pervasive and Mobile Computing journal. He has authored/edited two books published by IGI Global, USA, one book published by CRC Press, USA, and one book with Wiley, U.K. He received "Best Researcher of the Year 2015 of COMSATS Wah" award in 2015. He received the certificate of appreciation, "Exemplary Editor of the IEEE Communications Surveys and Tutorials for the year 2015" from the IEEE Communications Society. He received Best Paper Award from IEEE ComSoc Technical Committee on Communications Systems Integration and Modeling (CSIM), in IEEE ICC 2017. He consecutively received research productivity award in 2016-17 and also ranked# 1 in all Engineering disciplines from Pakistan Council for Science and Technology (PCST), Government of Pakistan. He also received Best Paper Award in 2017 from Higher Education Commission (HEC), Government of Pakistan.



Noel Crespi olds Masters degrees from the Universities of Orsay (Paris 11) and Kent (UK), a diplome d'ingenieur from Telecom ParisTech, a Ph.D and an Habilitation from Paris VI University (Paris-Sorbonne). From 1993 he worked at CLIP, Bouygues Telecom and then at Orange Labs in 1995. He took leading roles in the creation of new services with the successful conception and launch of Orange prepaid service, and in standardisation (from rapporteurship of IN standard to coordination of all mobile standards activities for Orange). In 1999, he joined Nortel Networks as telephony program manager, architecting core network products for EMEA region. He joined Institut Mines-Telecom in 2002 and is currently professor and Program Director, leading the Service Architecture Lab. He coordinates the standardisation activities for Institut Mines-Telecom at ITU-T, ETSI and 3GPP. He is also an adjunct professor at KAIST, an affiliate professor at Concordia University, and gust researcher at the University of Goettingen. He is the scientific director the French-Korean laboratory ILLUMINE. His current research interests are in Service Architectures, Sofwarization, Social Networks, and Internet of Things/Services. http://noelcrespi.wp.tem-tsp.eu/

Roberto Minerva holds a Ph.D in Computer Science and Telecommunications from Telecom SudParis, France. He was the Chairman of the IEEE IoT Initiative, an effort to nurture a technical community and to foster research in IoT. Roberto has been for several years in TIMLab with responsibilities on service architectures. Currently he is involved in activities on SDN/NFV (technical leader of the SoftFIRE H2020 project),

5G, Big Data, architectures for IoT. Now he is a research engineer in Telecom SudParis working on IoT software architecture and the digitalization of businesses in several industries. He is author of several papers published in international conferences, books and magazines.

How to cite this article: Saleem Y., Rehmani M. H., Crespi N., and Minerva R. (2019), Parking Recommender System Privacy Preservation through Anonymization and Differential Privacy, *Engineering Reports*, *xxx.*