applied sciences

MDPI

Article

Efficiency Boosts in Human Mobility Data Privacy Risk Assessment: Advancements within the PRUDEnce Framework

Fernanda O. Gomes [1,2,*], Roberto Pellungrini [3], Anna Monreale [2], Chiara Renso [4] and Jean E. Martina [1]

[1] Graduate Program on Computer Science, Department of Informatics and Statistics, Federal University of Santa Catarina (UFSC), Florianópolis 88040-370, SC, Brazil; jean.martina@ufsc.br
[2] Department of Computer Science, University of Pisa, 56126 Pisa, Italy; anna.monreale@di.unipi.it
[3] Classe di Scienze—Scuola Normale Superiore, 56126 Pisa, Italy; roberto.pellungrini@sns.it
[4] The Institute of Information Science and Technologies (ISTI) of the National Research Council (CNR), 56124 Pisa, Italy; chiara.renso@isti.cnr.it
* Correspondence: fernanda.gomes@posgrad.ufsc.br or f.oliveiragomes@phd.unipi.it

Abstract: With the exponential growth of mobility data generated by IoT, social networks, and mobile devices, there is a pressing need to address privacy concerns. Our work proposes methods to reduce the computation of privacy risk evaluation on mobility datasets, focusing on reducing background knowledge configurations and matching functions, and enhancing code performance. Leveraging the unique characteristics of trajectory data, we aim to minimize the size of combination sets and directly evaluate risk for trajectories with distinct values. Additionally, we optimize efficiency by storing essential information in memory to eliminate unnecessary computations. These approaches offer a more efficient and effective means of identifying and addressing privacy risks associated with diverse mobility datasets.

Keywords: privacy; privacy risk; privacy risk assessment; mobility; re-identification; computation improvements; risk; trajectory

check for updates

Citation: Gomes, F.O.; Pellungrini, R.; Monreale, A.; Renso, C.; Martina, J.E. Efficiency Boosts in Human Mobility Data Privacy Risk Assessment: Advancements within the PRUDEnce Framework. *Appl. Sci.* **2024**, *14*, 8014. https://doi.org/10.3390/app14178014

Academic Editors: Wenjie Zhang and Zhengyi Yang

Received: 24 July 2024
Revised: 3 September 2024
Accepted: 6 September 2024
Published: 7 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The extensive use of mobile devices equipped with location-tracking technologies, such as GPS, has collected a great deal of location data, offering insights into users' movements over time and space. A trajectory, in its raw form, consists of a sequence of spatio-temporal points that reveal the position of an object at specific times. Trajectories offer valuable information about human mobility patterns, benefiting various sectors such as security, urban planning, public transportation management, and disease prevention. However, using trajectory data also raises significant privacy concerns during data collection and sharing.

Individuals using these technologies face significant risks due to potential data breaches that can result in privacy violations. The collected data contain highly sensitive and personal information, making it vulnerable to re-identification attacks. These attacks aim to identify individuals or locations within trajectory datasets, posing a substantial threat to privacy. A privacy assessment study in the context of mobility data indicates that merely four spatio-temporal points can re-identify 95% of individuals in a low-granularity trajectory dataset [1]. Notably, the top three locations in a path are sufficient to identify over 80% of individuals [2]. The disclosure of location data raises significant privacy concerns, as it can be used to make intrusive inferences about individuals' habits, social behavior, and even religious and sexual preferences [3].

Privacy risk assessment is a process aimed at understanding which individuals in the data are at risk of privacy violations and quantifying the associated risk level. In Europe, the *General Data Protection Regulation (GDPR)*, and, similarly, in other countries such as with the *Lei Geral de Proteção de Dados (LGPD)* in Brazil, establishes principles and requirements

for the processing of personal data. These laws assign data controllers and processors to handle data, ensuring data protection. One important step for data custodians is to perform a quantitative privacy risk data assessment. Numerous privacy risk assessment methodologies based on probability and frequency have been proposed for evaluating the privacy risk across various data types [4–8].

One of the challenges in privacy risk assessment, especially concerning re-identification, is the need to reduce the computational resources required for evaluating privacy risk. One of the most accurate methods for privacy risk assessment is the simulation of background knowledge-based attacks. When simulating this type of attack, generating the background knowledge representing the adversary's knowledge about its victims is a particularly complex task, requiring substantial computational resources. As the adversary's knowledge expands, the computational complexity increases exponentially until half of the maximum possible knowledge size. Previous studies, such as those by Pellungrini et al. [9] and Naretto et al. [10], have explored using machine learning algorithms to mitigate this computational load, particularly when new data become available. Both methods were successful in significantly reducing the overall processing time. However, an initial computation of risk using combinations remains essential to establish the training dataset for their approach, which means that the lengthy execution time is still a challenge that needs to be addressed.

Due to the data size, the process becomes more complex when dealing with trajectory data. Background knowledge related to trajectories often consists of sequences of visited places or simply locations, representing places or visits the attacker is aware of regarding the victim's movement. As this knowledge grows, the number of potential background knowledge configurations to evaluate increases significantly. Each configuration requires risk assessment, further adding to the overall complexity of the process. The time required for this assessment depends on the dataset size, number of trajectories, and trajectory length.

This work addresses computational challenges encountered in risk assessment analysis using PRUDEnce [11], a state-of-the-art privacy risk assessment framework for background knowledge-based attacks. We aim to develop strategies to mitigate the computational complexity of evaluating privacy risks in trajectory datasets. We explore the complexities arising from the high computational demands of re-identification risk assessment. We propose different computational improvements and optimization strategies to simplify the risk assessment process, enhance computational efficiency, and facilitate more scalable and accurate analyses.

Our contributions include validating the significance of low entropy and volatile feature frequency to reduce computational complexity in re-identification risk assessment. We explain how these factors impact re-identification risk and explore methodologies for taking advantage of low-entropy characteristics to simplify risk assessment processes. Additionally, we introduce optimization strategies to enhance computational efficiency in re-identification risk assessment. These strategies include the avoidance of redundant computations by storing background knowledge configurations and the optimization of memory usage through the utilization of unique values.

Furthermore, we provide a thorough analysis of the results obtained from implementing the proposed computational enhancements and optimization techniques, demonstrating significant reductions in complexity. We also demonstrate how our proposed optimizations can effectively reduce the execution time of the risk assessment process. This improvement is particularly beneficial as it enhances the efficiency of the re-identification risk assessment, allowing for faster trajectory data processing. By reducing execution time, our optimizations contribute to improved scalability and usability of the risk assessment methodology, making it more practical for real-world applications.

The paper is organized as follows. Section 2 presents the state of the art of privacy risk assessment frameworks. Section 3 provides the data definitions regarding trajectories, attacks, privacy risk assessment, and combination complexity. Section 5 presents the

optimization techniques. Section 6 shows the experimental details and the results with a final discussion. We conclude and plan future work in Section 7.

2. Related Work

Quantitative privacy risk assessment for mobility trajectory and other types of data is a well-studied topic. Trabelsi et al.'s (2009) [4] approach involves recommending secure configurations through a smart bootstrapping system, aiming to enhance understanding and management of the risks associated with non-controlled data disclosure. The authors utilized a probability-based approach, demonstrating that it is possible to reduce computation time by leveraging previously calculated risk values to predict future risk values. Song et al. (2014) [6] propose a modification-based anonymization approach and evaluate privacy risk based on the uniqueness of trajectory data. The authors employed a probability of re-identification based on sub-trajectories and demonstrated that, by reducing the overall trajectory size—specifically, by removing the highest-risk sub-trajectories—the re-identification risk is significantly decreased. In Achara et al. (2015) [5], their research investigates the privacy implications of the list of apps installed by users on smartphones, emphasizing the re-identifiability issue. Analyzing a dataset with 54,893 Android users over 7 months, the study finds that merely four installed apps are sufficient for user re-identification over 95% of the time. Remarkably, the complete list of installed apps is unique for 99% of users, making it susceptible to tracking or profiling by services like Twitter with access to this information. In [12], their proposed framework integrates runtime risk assessment into information disclosure access control, utilizing disclosure risk for decision-making. Access-control decisions are driven by the associated disclosure risk of data access requests, and adaptive anonymization serves as a method for mitigating risks, ensuring privacy preservation. Other studies in the literature explore re-identification risk as a privacy measure within the realms of network and social media data [7,8].

In [7], the authors introduce a framework for assessing privacy and anonymity within social networks and introduce a new re-identification algorithm aimed at anonymized social network graphs. To demonstrate its effectiveness on real-world networks, they showed that a third of users with accounts on both Twitter and Flickr can be re-identified in the anonymized Twitter graph with a 12% error rate. Finally, ref. [8] showed that, based on social media behavior, it is possible to re-identify passive web visits to the host. Their method combines a public follower graph on social media with posting behaviors and time-based inferences and proved to be efficient in re-identifying the users. Khalfoun et al. (2021) [13] propose EDEN, selecting optimal Location Privacy Protection Mechanisms using federated learning without exposing raw traces, demonstrating superior privacy vs. utility tradeoff across real-world datasets. Silva et al. (2022) [14] introduce the Personal Data Analyser, which employs automated data monitoring with Regular Expressions, NLP, and machine learning to enhance privacy. Integrated into the PoSeID-on platform, it alerts users to risks with crisp and fuzzy models validated through real-world use cases.

In this work, we adopted the PRUDEnce framework introduced by Pratesi et al., as elucidated in their seminal work [11] and previously presented in Section 6. PRUDEnce was introduced as a system that deals with finding a balance between privacy risk and data usefulness when sharing sensitive human activity data. This framework offers a methodology for the computation of privacy risk in a data-driven fashion. At its essence, PRUDEnce revolves around the foundational principle of k-anonymity, wherein the privacy risk assessment is linked to the dimensions of k-sets associated with each individual in the dataset. The method checks out real privacy risks for users and ensures data quality for those not at risk. Data providers can try different changes to strike the right balance between privacy and usefulness. The practical effectiveness of PRUDEnce is shown with real mobility data, exploring presence, trajectory, and road segment data formats. Our decision to utilize PRUDEnce was based on its flexible extension and suitability for trajectory data.

The computational intensity of PRUDEnce has prompted exploration into machine learning approaches aimed at predicting privacy risk, thereby bypassing the need for

computationally exhaustive processes. Pellungrini et al. (2017) [9] present a swift and adaptable method for estimating privacy risk in human mobility data. Their approach involves training classifiers to link individual mobility patterns with different privacy risk levels. Another important advancement in this field is the EXPERT framework, developed by Naretto et al. (2020) [10]. This framework refines PRUDEnce by introducing a machine learning methodology that proficiently forecasts privacy risk from sequential data. Moreover, the framework enhances the interpretability of these predictions by incorporating methodologies such as SHAP [15] and LIME [16]. Another study proposed by Naretto et al. (2023) [17] presents an optimization of EXPERT, the EXPHLOT. Authors use distinct time series classifications, such as ROCKET and INCEPTIONTIME, to improve risk prediction while reducing computation time.

While previous works focus on improving efficiency and reducing computational demands for privacy risk assessment, these works often require an initial conventional risk analysis to generate training data for the risk-predicting machine learning model. Our proposal offers an approach that aims to enhance the computational risk algorithm. By directly optimizing the risk assessment process, our methodology eliminates unnecessary computation. This streamlined approach not only reduces the computational time but also simplifies the overall risk assessment pipeline.

Our strategy for evaluating the maximum risk and reducing the computation time is to select low-entropy trajectory features to target high-risk data and reduce the data used to evaluate the risk. Pellungrini et al. (2017) [9] showed that entropy has an important impact on predicting features/locations in machine learning models. The idea is that location entropy is related to uniqueness, which is the main measure of anonymity. If a user passes through high-entropy locations, where, therefore, many different people pass through, the uniqueness of their mobility profile is lost as the general movement blurs it. In EXPHLOT [17], authors show that they have the highest entropy locations, evaluating only the lowest entropy locations. In this way, they focus on locations with fewer individuals visiting, focusing on explaining high-risk predictions. In our work, we reduced the computation time for maximum risk evaluation since it would yield the highest risk values. We also went a step beyond checking not only location entropy but also time entropy. Using different attacks and adversary knowledge sizes, we used the p-value and Kolmogorov–Smirnov test [18,19] to prove the efficiency of using low-entropy values to reduce maximum risk computation.

Our proposal introduces novel techniques to more efficiently identify and prioritize high-risk trajectories. By leveraging insights from trajectory data characteristics, such as inherent uniqueness and temporal dependencies, our algorithm can highlight trajectories with elevated privacy risks.

3. Basic Concepts

This section provides an overview of the fundamental concepts related to trajectories and location attacks. We introduce the PRUDEnce framework, and discuss the computational complexity of combinations.

3.1. Trajectory

A trajectory, also known as a raw trajectory, is a sequence of spatio-temporal points, defined in Definition 1. Each point, detailed in Definition 2, includes spatial coordinates and a timestamp, referred to as trajectory features in this work. A segment of a trajectory is called a sub-trajectory, as described in Definition 3, which can also be considered a trajectory.

Definition 1 (Trajectory). *A trajectory T is a sequence of spatio-temporal points $T = (p_0\langle x_0, y_0, t_0\rangle, \ldots, p_n\langle x_n, y_n, t_n\rangle)$, where (x_i, y_i) are spatial coordinates, and t_i represents time, with $t_0 < t_1 < \ldots < t_n$ to maintain chronological order.*

Definition 2 (Point). *A point p is a tuple $\langle x, y, t \rangle$, where x and y are spatial coordinates representing a location, and t is the time of the visit.*

Definition 3 (Sub-trajectory). *A sub-trajectory s of a trajectory T is an ordered sequence of points from T, defined as $s = (p_{i_1}, p_{i_2}, \ldots, p_{i_k})$, where s contains at least one point but fewer than all points of T.*

In this work, we use the terms *point* or *visit* to refer to a single element of a trajectory, while, by the term *location* l, we refer to the point's spatial information. A subsequence of locations (Definition 4) is an ordered list of locations. We denote by $U_{set} = \{u_1, \ldots, u_n\}$ the set of the distinct individuals represented in the mobility dataset D, formally described in Definition 5.

We use, in this work, the terms *point* or *visit* refer to an individual element of a trajectory, while the term *location* l specifically denotes the spatial aspect of a point. A subsequence of locations (Definition 4) is defined as an ordered sequence of locations. The set $U_{set} = u_1, \ldots, u_n$ represents the distinct individuals captured in the mobility dataset D, as outlined in Definition 5.

Definition 4 (Subsequence). *Let $\mathcal{L} = \{l_1, l_2, \ldots, l_w\}$ represent a set of locations. A sequence $S = \langle s_1, s_2, \ldots, s_m \rangle$ is an ordered list of locations from $\mathcal{L}$, where each location can appear multiple times.*

A sequence $T = \langle t_1, t_2, \ldots, t_z \rangle$ is called a subsequence of S (denoted $T \preccurlyeq S$) if there are indices $1 \leq i_1 < i_2 < \cdots < i_z \leq m$ such that

$$t_j = s_{i_j} \quad \textit{for} \quad j = 1, 2, \ldots, z.$$

This ensures that T maintains the order of S.

Definition 5 (Mobility Dataset). *A mobility dataset D is a collection of trajectories, $D = \{T_1, T_2, \ldots, T_n\}$, where each T_u represents the trajectory of a moving object u $(1 \leq u \leq n)$. For multiple-aspect trajectories, the dataset is represented as $D = \{MAT_1, MAT_2, \ldots, MAT_n\}$.*

3.2. Risk of Re-Identification

Re-identification happens when an adversary successfully links the anonymized or otherwise protected data of an individual with information available to them, whether obtained publicly or through other means. In [20], the authors of the paper comprehensively review terminology and the methodologies related to the risk of re-identification. There are two principal manners to evaluate the re-identification risk: at the dataset and individual levels. The dataset risk involves the proportion of records an adversary can re-identify from a protected dataset. Our work focuses on reducing the computation time of individual risk assessment.

The re-identification risk of an individual is articulated as the probability that a particular sample record of an adversary is identified as corresponding to a specific individual in the dataset, influenced by the observation that risk exhibits non-uniformity across the dataset, with rare combinations of sensitive attributes potentially leading to the re-identification of individuals [21]. As defined in [22], where there are k possible combinations of key attributes inducing a partition, the individual disclosure risk for a record with the k-th combination is inversely proportional to the known population frequency F_k, expressed as $\frac{1}{F_k}$. In both risk measures, adversaries commonly employ the primary *Data Matching* technique. This method centers on establishing connections between records, aiming to identify those belonging to the same individual across different databases. Our work aims to reduce the complexity of the re-identification risk computation within the PRUDEnce framework.

3.3. PRUDEnce Framework

PRUDEnce, a privacy risk assessment framework proposed by [11], is recognized for its effectiveness in assessing privacy risks in trajectory data. It plays a crucial role in helping data providers (DPs) make informed decisions while maintaining data quality. The framework is designed to support GDPR compliance, with a particular focus on Article 25, which emphasizes data protection by design and default, aligning with the principles of Data Protection Impact Assessments. PRUDEnce is not limited to a specific country or jurisdiction, but it is primarily structured to comply with GDPR (European Union). However, due to its adaptable nature, PRUDEnce can be extended to assess privacy risks in other legal contexts. PRUDEnce provides a universal methodology for privacy risk assessment in any type of data.

PRUDEnce assesses privacy risks in data sharing, particularly focusing on empirical privacy risks during the transfer of raw personal data from the DP to the Service Developer (SD). Background knowledge dimensions are crucial for evaluating potential privacy risks and outlining external information accessible to potential attackers. This context-dependent background knowledge impacts the effectiveness of privacy attacks.

Background knowledge represents an adversary's knowledge subset regarding a user u. The methodology evaluates privacy risks with varying levels of background knowledge, from minimal to maximal knowledge, enabling responsible decision-making. Defining attack models and background knowledge is critical, systematically balancing privacy risks and data utility.

A *background knowledge category* refers to information known by an adversary about the specific dimensions of an individual's data. For instance, in mobility data, typical dimensions include space, time, frequency of visiting a location, and probability of visiting a location. The number of elements the adversary knows, the size of the *background knowledge configuration*, is denoted by k. An example of a background knowledge configuration could be the adversary knowing $k = 3$ points in the trajectory of an individual. An *instance of background knowledge* represents specific information the adversary knows, such as a visit to a specific location; we can see an example of background knowledge in Figure 1. In Figure 1, we begin with a dataset containing trajectory data. In the example provided, the attacker has knowledge of location sequences, with a knowledge size of 2. The second table in Figure 1 illustrates the background knowledge instances generated from the input dataset. Location information is combined in pairs (two by two) to represent the potential knowledge an attacker might possess.

These concepts are formalized as follows, according [11], in Definition 6.

Definition 6 (**Background Knowledge Category, Configuration and Instance**). *We consider a background knowledge category, denoted as $\mathcal{B}$. Within this category, we define B_k as a specific configuration of background knowledge, where B_k belongs to the set $\mathcal{B} = B_1, B_2, \ldots, B_n$. The value of k indicates the number of elements within $\mathcal{B}$ that an adversary possesses. Each individual element b that is part of B_k represents a distinct instance of this background knowledge configuration.*

$$B_k = \left\{ \binom{\{value_1, \ldots, value_{n(u)}\}}{k} \mid \forall\, u \right\}$$

Let $\mathcal{D}$ be a database, D a dataset extracted from $\mathcal{D}$ as an aggregation of the data on specific dimensions (e.g., an aggregated data structure and filtering on some dimension), and D_u and D_u the subset of records corresponding to individual u within D; we establish the likelihood of re-identification as follows in Definition 7 and in 2 in Figure 1.

Definition 7 (**Probability of re-identification**). *Given an attack, we consider a function $matching(d, b)$ that determines whether a record $d \in D$ corresponds to the background knowledge instance $b \in B_k$. We then define a function $M(D, b) = \{d \in D \mid matching(d, b) = True\}$, which*

identifies all records in D that match b. The probability of re-identification for an individual u within the dataset D is expressed as

$$PR_D(d = u \mid b) = \frac{1}{|M(D,b)|}$$

which represents the chance of linking a record $d \in D$ to an individual u, given the instance $b \in B_k$.

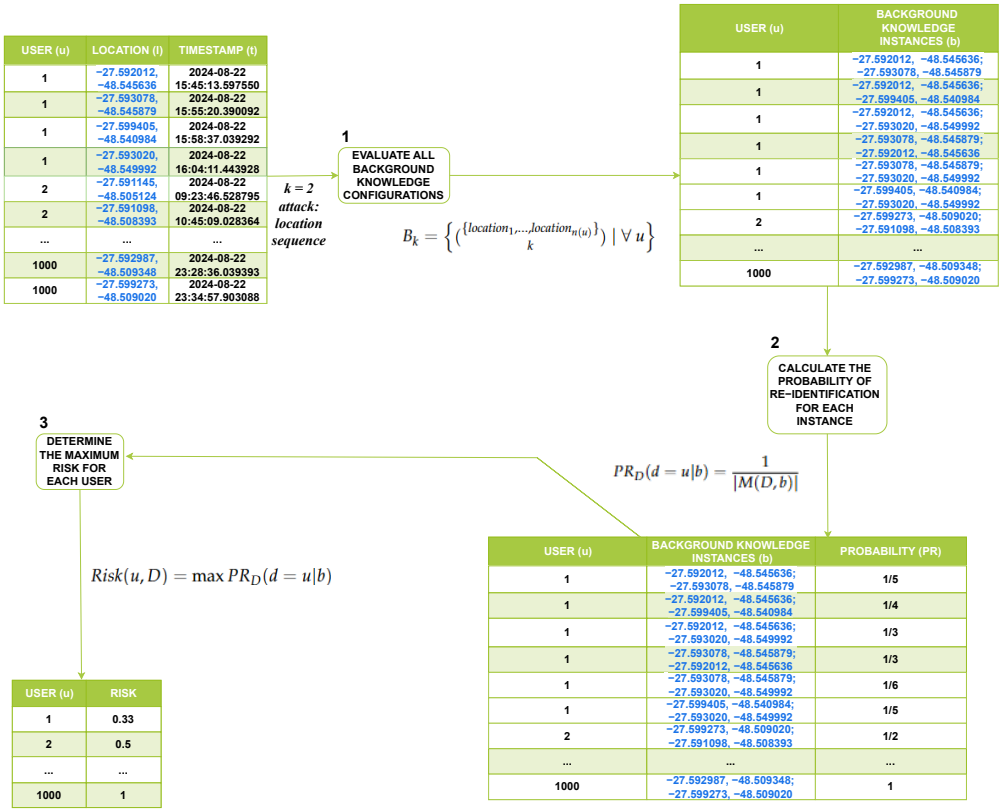

Figure 1. Data flow in the privacy risk assessment within PRUDEnce.

The *compatibility* is expressed by a function $matching(d, b)$, which indicates whether or not a record $d \in D$ matches the instance b. The matching function depends on the background knowledge used during the attack. The PRUDEnce framework characterizes the re-identification risk for an individual as the maximum probability of re-identification among all instances within a background knowledge configuration, as defined in Definition 8 and shown in Step 3 of Figure 1.

Definition 8 (Re-identification Risk or Privacy Risk). *The re-identification risk, or privacy risk, for a specific individual u, associated with a background knowledge configuration B_k, is determined as the highest probability of re-identification, expressed as $Risk(u, D) = \max PR_D(d = u \mid b)$, where $b \in B_k$. This risk is bounded by a minimum threshold of $\frac{|D_u|}{|D|}$, which corresponds to a random guess within dataset D, and is $Risk(u, D) = 0$ when u is not present in D.*

An individual may face various privacy risks, each corresponding to different configurations of background knowledge used in an attack. Initially, an attack is formulated and customized to use a specific category of background knowledge. Subsequently, a range of background knowledge configurations is examined, denoted as $\{B_1, \ldots, B_m\}$. For each configuration B_k, all instances b within B_k are analyzed, along with their respective probabilities of re-identification. Finally, the maximum probability of re-identification

across all instances b within configuration B_k determines the privacy risk for the individual in that specific context.

3.4. Location Attacks

In trajectory datasets, a "location attack" involves determining sensitive information about individuals by analyzing their movement patterns and the locations they visit.

Various attacks have been proposed in the literature to exploit such vulnerabilities. For instance, the location sequence attack, as outlined in works such as [9,23], involves adversaries possessing knowledge of a subset of the locations visited by an individual and the temporal sequence of these visits. Similarly, the visit attack, outlined in studies like [3,24,25], requires adversaries to be privy to information about a subset of the locations visited by an individual along with the specific times of these visits. For example, in a trajectory dataset containing GPS coordinates or timestamps of individuals' movements, a location attack could involve analyzing this data to identify specific places individuals frequently visit, such as their homes, workplaces, or other sensitive locations. By correlating these patterns with additional information, such as social media posts or public records, adversaries may be able to deduce private details about individuals, such as their daily routines, habits, or interests.

3.4.1. Location Sequence Attack

In the location sequence attack, introduced in [9,23], the an adversary is aware of a subset of the locations that the individual has visited and the temporal ordering of the visits.

In the context of trajectory data privacy, a location sequence attack involves an adversary possessing knowledge of a subset of the locations visited by an individual, as well as the temporal ordering of these visits. For an individual s, the sequence of visited locations $L(T_s)$ is represented by the sequence of locations l_i within T_s.

The background knowledge category for a location sequence attack is formally defined as follows.

Definition 9 (Location Sequence Background Knowledge). *Let k be the number of locations l_i known by the adversary for an individual s. The location sequence background knowledge comprises configurations based on k locations, denoted as $B_k = L(T_s)[k]$. Here, $L(T_s)[k]$ represents the set of all possible k-subsequences of the elements in set $L(T_s)$.*

In this context, the notation $a \preccurlyeq b$ indicates that a is a subsequence of b. Each instance $b \in B_k$ is thus a subsequence of location $X_s \preccurlyeq L(T_s)$ of length k. Given a record d in the dataset D and the corresponding individual u, the matching function is defined to determine the presence of a location sequence:

$$evaluate(T, b) = \begin{cases} 1, & b \preccurlyeq_l L(T_u)^k \\ 0, & otherwise \end{cases} \quad (1)$$

In this attack, the attacker knows that a person went, first, to a supermarket and then to work, but they do not know when, only the sequence of places.

3.4.2. Visit Attack

In this attack, introduced in [3,9,24,25], an an adversary is aware of a subset of the locations that the individual has visited and the time the individual visited these locations. Let k be the number of visits vs. of an individual s known by the adversary. The visit background knowledge consists of configurations derived from k visits, formally represented as $B_k = T_s[k]$, where $T_s[k]$ indicates the set of all possible k-length subsequences within the trajectory T_s.

Each instance $b \in B_k$ represents a spatio-temporal subsequence X_s of length k. The subsequence X_s positively matches a given trajectory if the trajectory aligns with b in both

spatial and temporal aspects. Formally, given a record d in the dataset D, the matching function is defined as

$$evaluate(T,b) = \begin{cases} 1, & \forall (l_i, t_i) \in b, \exists (l_{d_i}, t_{d_i}) \in d \text{ such that } l_i = l_{d_i} \wedge t_i = t_{d_i} \\ 0, & otherwise \end{cases} \quad (2)$$

In this attack, the attacker knows that, for instance, a person went, first, to a supermarket at 1 p.m. and after work at 2 p.m. They know the time when the person visited the place.

In the next subsection, we will introduce the *combinatorial problem*.

4. Computational Complexity of Combinations

In addressing computational challenges, particularly in evaluating re-identification risk, we encountered significant memory and complexity issues. These challenges primarily arise from the high computational complexity of the risk evaluation process, which is denoted by $\mathcal{O}(\binom{len}{k} \times N)$. Here, $\binom{len}{k}$ represents the generation of background knowledge configuration sets, where len indicates the size of the trajectory, and N denotes the number of matching operations required for each configuration. This complexity, as highlighted in [9,11], poses substantial difficulties, especially in scenarios involving empirical privacy risk.

Re-identification risk via a background knowledge attack simulation requires analyzing the likelihood of identifying a specific user within a dataset, considering various types of background knowledge for potential adversaries. However, the computational complexity grows exponentially with the size of the trajectory and the number of potential background knowledge instances.

The computational complexity of combinations denoted as $\binom{n}{k}$ can be analyzed in terms of factorials and depends on the values of n and k. The formula for combinations is given by

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

The Binomial Coefficient Function

The binomial coefficient denotes the number of ways to choose k outcomes without considering their order from a total of n possibilities. This concept is commonly recognized as a combination. Figure 2 shows the binomial coefficient behavior according to n and k values. Some important characteristics can be noticed.

The binomial coefficient curve has several key characteristics. It exhibits symmetry around its peak due to the symmetry in the combinations formula $C(n,k) = C(n,n-k)$, where choosing k elements is equivalent to choosing $n-k$ elements. Consequently, the curve is symmetric around the middle point. The curve starts at 1 when $k = 0$ (choosing 0 elements) and ends at 1 when $k = n$ (choosing all elements), reflecting the fact that there is only one way to choose 0 elements (no choice) and one way to choose all n elements (all elements are chosen).

The peak value of the curve occurs at the middle point, where $k = \frac{n}{2}$ (rounded up or down depending on whether n is even or odd). The number of ways to choose k elements is maximized, leading to the highest binomial coefficient. However, the binomial coefficient gradually decreases as k deviates from the middle point. This is because choosing fewer elements as it moves away from the middle results in a decrease in the number of combinations.

The binomial coefficient specifically represents combinations that do not consider the order of elements. In contrast, permutations, where order matters, would result in a different curve behavior. From a computational perspective, factorial computation can be computationally expensive, especially for large values of n and k, leading to large

intermediate values. However, optimizations can be applied to enhance efficiency in computing binomial coefficients.

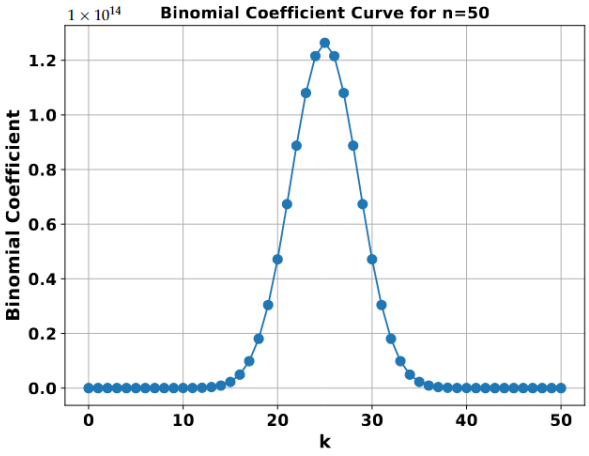

Figure 2. Behavior 1×10^{14} of binomial coefficient curve.

5. Privacy Risk Assessment: Advancements within the PRUDEnce Framework

This section proposes improvements within the PRUDEnce framework, particularly on computing privacy risk assessment. As illustrated in Figure 3, we introduce several advancements: Low Entropy, Cache Strategy, Break, Direct Evaluation, and Reuse. Each of these techniques targets a specific aspect of the privacy risk assessment process. The Low Entropy approach reduces the number of instances that require re-identification probability evaluation. The Cache Strategy stores instance information in memory, minimizing redundant calculations. The Break method stops the evaluation once an instance with maximum risk is identified. Direct Evaluation bypasses risk computation when the trajectory contains any feature with unique information. Finally, the Reuse technique applies the privacy risk evaluation from the $k-1$ analysis to the current k analysis if the previous risk is already at its maximum.

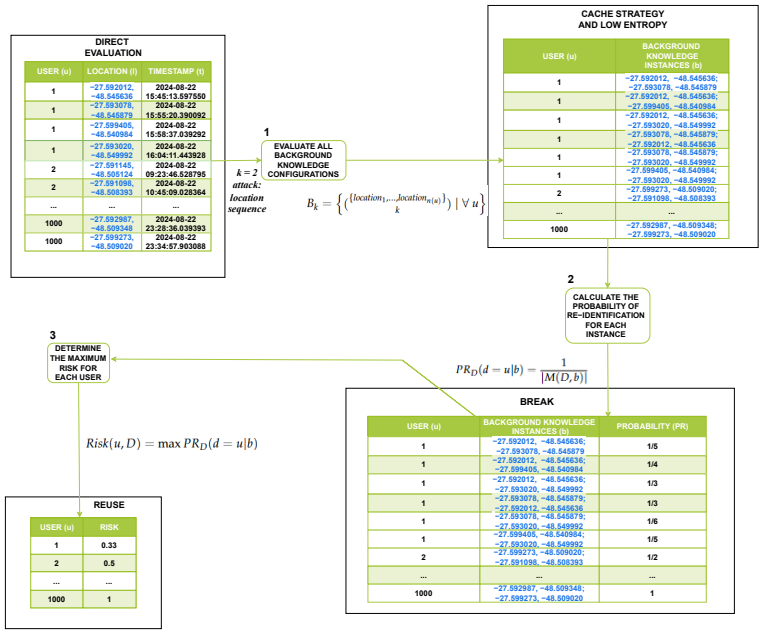

Figure 3. Privacy risk data flow and strategies relation.

5.1. Relationship between Entropy, Feature Frequency, and Re-Identification Risk

When analyzing data with entropy-applied trajectory feature frequency, such as location or time, it becomes clear that features with lower entropy values also have lower frequencies and pose a higher risk of re-identification for the individuals. This relationship between entropy and frequency highlights several key factors contributing to the increased risk.

Less frequent features tend to have a lesser impact on the overall uniqueness of the data. With fewer instances, each occurrence becomes more significant in distinguishing individuals within the dataset. Consequently, the aggregated data offer less anonymity, making it easier for adversaries to differentiate between individuals.

Regarding entropy, low-frequency features contribute less to the overall uncertainty in the dataset because their probability of occurrence is low. In entropy calculations, probabilities of rare events (e.g., visits to low-frequency locations) have less impact on the overall entropy than probabilities of more common events. Therefore, low-frequency features tend to have lower entropy, indicating less uncertainty in the distribution of visits. While low-frequency features may be easier to predict due to their less variable and more predictable nature, this predictability can increase the risk of re-identification.

Applying entropy to features highlights the heightened risk associated with low-frequency data. These features, characterized by their low entropy, offer less anonymity and increased predictability, making them more susceptible to re-identification.

5.1.1. Formal Proof

Here, we present formal proof regarding the correlation between the entropy of features' frequency distribution and the risk of re-identification.

Claim: trajectory features with low entropy in their frequency distribution are those with low frequency, and they are more at risk of re-identifying the data owner.

Proof. entropy is a measure of uncertainty or randomness in a probability distribution. Let x_{ui} represent the i-th feature of record x_u of dataset $D_{n=1}^{N}$ where N is the total number of individuals. The Shannon Entropy [26] for feature x_i is given by

$$E(x_i) = -\sum_{u=0}^{N} p_u(x_{ui} = v) \log_2 p_u(x_{ui} = v)$$

where $p_u(x_{ui} = v)$ is the probability that individual u has a feature value $vs.$ for x_i

The probability $p_u(x_{ui} = v)$ can be calculated based on its frequency:

$$FR(u, v) = |\{x_j i | x_j i = vs.j = u\}|$$

in individual u's data divided by the individual data size N:

$$p_u(x_{ui} = v) = \frac{FR(u, v)}{N}$$

Substituting $p_u(x_{ui} = v) = \frac{FR(u,v)}{N}$ into the entropy formula, we obtain

$$E(x_i) = -\sum_{u=0}^{N} \frac{FR(u, v)}{N} \log_2 \frac{FR(u, v)}{N}$$

For a feature with low frequency, $FR(u, v)$ is small for all $u = 0, \ldots, N$. Therefore, the probability $p_u(x_{ui} = v)$ is also small. In the low-frequency scenario, where the feature value frequency is low, the probability $p_u(x_{ui} = v)$ is also low, leading to a smaller contribution to the overall entropy. This is because $p_u(x_{ui} = v) \log_2(p_u(x_{ui} = v))$ is close to zero when $p_u(x_{ui} = v)$ is small. Hence, the entropy $E(x_i)$ is lower for low-frequency features.

Conversely, in the high-frequency scenario, where the feature value frequency is high, the probability $p_u(x_{ui} = v)$ is higher. This results in a larger contribution to the overall

entropy, as $p_u(x_{ui} = v) \log_2(p_u(x_{ui} = v))$ is higher when $p_u(x_{ui} = v)$ is larger. Hence, the entropy $E(x_i)$ is higher for high-frequency features.

Base Case: we consider the case where *vs*. appears only once in the dataset. This means that $FR(u, v) = 1$ for an individual u and 0 for all others. The entropy calculation becomes

$$E(x_i) = -\left(\frac{1}{N}\log_2\frac{1}{N}\right) = -\left(\frac{1}{N}\cdot(-\log_2 N)\right) = \frac{\log_2 N}{N}$$

Since $S(u)$ is a finite positive number, $\frac{\log_2 N}{N}$ is positive but small. Thus, $E(x_i)$ is low.

Inductive Step: we assume that, for a feature value *vs*. appearing k times in the dataset, the entropy $E(x_i)$ is low. We must show that the entropy remains low if the feature x_i appears $k + 1$ times.

If *vs*. appears $k + 1$ times, the probabilities $p_u(x_{ui} = v)$ will still be small because $FR(u, v)$ divided by N remains small. Thus, the term $p_u(x_{ui} = v) \log_2(p_u(x_{ui} = v))$ for each $u = 0, \ldots, N$ will contribute a small value to the overall entropy.

With these small contributions, we see that the entropy $E(x_i)$ will increase slightly but remain low because the additional term for the $k + 1$-th occurrence is small.

Therefore, by induction, features with low frequency have low entropy.

Conclusion: features with low entropy in their frequency distribution, indicating low frequency, contribute less to the overall uncertainty. However, this also means that these features are more unique. The uniqueness of these features makes it easier to re-identify the data owners, as fewer individuals have these low-frequency features. This establishes a link between local entropy and uniqueness in the data. Therefore, we have shown that features with low entropy are indeed those with low frequency, and can pose a higher risk of re-identifying the data owner. For this reason, we use entropy and frequency to reduce the size of the background knowledge set when evaluating maximum risk and *empirical privacy risk* across the entire dataset. □

5.1.2. Selecting Background Knowledge Configurations with Low-Entropy Trajectory Features Frequency

The approach aims to enhance the identification of background knowledge instances with heightened privacy risks in the dataset. It involves mathematical definitions and procedures to systematically identify and retain instances that meet the specified criterion.

Definition 10 (Low-entropy Feature Set). *Let x_{ui} represent the i-th feature of record x_u of dataset $D_{n=1}^N$ where N is the number of individuals, and let $E(x_i)$ denote the entropy associated with feature value $x_{ui} \in D_{n=1}^N$. We define the set of low-entropy features $X_{low\text{-}entropy}$ as*

$$X_{low\text{-}entropy} = \{x_{ui} \in D_{n=1}^N \mid E(x_i) \leq percentile(E(x_i), y)\},$$

where $percentile(E(x_i), y)$ is the y-th percentile of the distribution of entropy values $E(x_i)$.

Definition 11 (Low-Entropy Selected Instances). *If B_k is a set of subsets of $D_{n=1}^N$, where each $b \in B_k$ contains feature $x_{ui} \in D_{n=1}^N$, then, for each $b_j \in B_k$ (where j represents the j-th instance), we define the selected background knowledge instances as*

$$b_j = \{x_{ui} \in b_j \mid x_{ui} \in X_{low\text{-}entropy}\}.$$

The set of all selected low-entropy subsets $B_{low\text{-}entropy}$ is defined as

$$B_{low\text{-}entropy} = \{b_j \mid b_j \in B_k\}.$$

In Definition 10, the low-entropy $X_{low-entropy}$ feature set is created, selecting only the feature values with the lowest values. All possible background knowledge configurations B_k are generated, but only instances containing the selected feature value in the y% lowest entropy results are selected for risk evaluation. Only the background knowledge instances

b that contain feature values from $X_{low-entropy}$ are retained for risk evaluation, as defined in Definition 11. This ensures that the analysis focuses on instances featuring feature frequency with low entropy, thereby improving privacy risk assessment by considering most relevant and distinctive features from that specific dataset.

Given their infrequency or uniqueness within the dataset, low-frequency features are prone to being distinctive or having limited occurrences. Consequently, background knowledge instances containing such features are more likely to contribute to higher risk values. We have the same or very similar maximum risk values as we compute all instances of risk. As a result, including infrequent feature values in the background knowledge raises the likelihood of identifying specific individuals.

This formal approach provides a systematic method for selecting background knowledge instances containing low-entropy feature frequency, thereby enhancing the identification of instances with heightened privacy risks in the dataset. By incorporating the reduced background knowledge configuration into the risk assessment framework, we can formally analyze how targeting low-entropy values in the background knowledge configuration can lead to computational improvements in privacy risk assessment. This approach allows us to focus computational efforts on configurations featuring low-entropy features, thereby facilitating risk assessment and enhancing the accuracy of re-identification risk estimates.

5.1.3. Complexity Analysis

Reducing the number of instances to evaluate will reduce the computational complexity. By reducing the number of instances, we decrease the number of matching operations, leading to lower time and resource requirements.

To assess the risk, we simulated an attack by calculating all possible k-combinations of information that an attacker might possess. For each combination of k points, we assumed that the attacker utilized all these points to carry out the attack. This resulted in a high computational complexity of $O\left(\binom{\text{len}}{k} \times N\right)$ because the framework created $\binom{\text{len}}{k}$ different configurations of background knowledge and, for each configuration, it performed N matching operations using the matching function. We supposed we could reduce the number of combinations. In that case, we would also reduce the number of matching operations, decreasing the value of N and lowering the overall complexity.

5.2. Optimizations

This section proposes optimization strategies to enhance computational efficiency in re-identification risk assessment. We discuss approaches such as avoiding redundant computation by saving background knowledge instances and utilizing unique values to optimize memory usage.

5.2.1. Cache Strategy

We saved computational resources in the **Cache** Strategy by precomputing and caching background knowledge instances. We leveraged this approach to optimize risk computation efficiency while maintaining the PRUDEnce framework's accuracy.

Saving combinations in memory offers advantages such as computational efficiency, time savings, resource conservation, scalability, dynamic updates, flexible risk analysis, and improved response time. This approach avoids redundant calculations, enables quick risk assessments for different instances, and conserves memory compared to recomputing combinations. Overall, storing combinations in memory optimizes the risk assessment processes.

The computation of privacy risks in trajectory data analysis often involves evaluating multiple background knowledge instances for each user trajectory. However, this process can be computationally intensive, especially with large datasets. We propose a novel strategy to address this challenge that reduces redundant computations by precomputing and caching background knowledge instances.

Our approach involves two main steps:

1. **Precomputation**: We generate and cache all background knowledge configurations for each user trajectory. This step is performed once and requires computational resources in advance, but it results in significant savings during subsequent risk computations.
2. **Reuse**: During risk computation, we retrieve precomputed instances' risk value from memory instead of generating background knowledge instances dynamically. This eliminates the need for redundant computations and reduces the overall computational overhead.

By implementing this approach, we aim to simplify the risk assessment process and improve the scalability of our trajectory data analysis.

The **Original** approach, proposed in PRUDEnce, involves dynamically computing background knowledge instances for each user trajectory during risk computation. This approach is straightforward but can be computationally inefficient, especially for large datasets, as shown in Algorithm 1.

Algorithm 1 Risk computation with original approach

```
for each user trajectory traj do
    Compute all possible background knowledge instances with size k
    for each background knowledge instance inst do
        Compute risk for inst using matching function and update maximum risk
    end for
end for
```

The proposed approach, named **Cache**, precomputes and caches background knowledge instances, reducing redundant computations and improving computational efficiency, as shown in Algorithm 2.

Algorithm 2 Risk computation with Cache approach

```
Precompute and cache all background knowledge instances
for each user trajectory traj do
    Compute all possible background knowledge instances with size k
    Compute the matching function for each instance and save the results in memory
    for each background knowledge instance inst do
        Compute risk for inst by accessing the matching value in memory
    end for
end for
```

In the **Original** approach, the complexity of computing risks for all trajectories is $O\left(\binom{\text{len}}{k} \times N\right)$, as each trajectory requires matching function overall background knowledge instances.

With the **Cache** strategy, the complexity is reduced to $O\left(\binom{\text{len}}{k}\right)$ for risk computation, since the matching needs to be evaluated only once and retrieving the information from memory is linear. Therefore, the overall complexity is significantly lower compared to the **Original** approach.

5.2.2. Unique Values and Direct Evaluation

While saving combinations in memory brings several advantages, it also presents challenges and considerations. One main issue is the potential for increased memory usage, especially when dealing with large datasets or many combinations. Storing all possible combinations can lead to high memory requirements, which may demand many system resources.

The scikit-mobility library in Python (https://scikit-mobility.github.io/scikit-mobility/) (accessed on 1 May 2023), introduces an improvement related to PRUDENce risk assessment and its computation, which we will call the **Break** approach. This improvement involves the

force_instances parameter. When using the **Original** approach and determining maximum risk, if *force_instances* is set to false and a single maximum value is detected, indicating a risk value of one, there is no need to assess additional background knowledge instances, as can be seen in Algorithm 3. This is because the trajectory risk is already at its maximum.

Algorithm 3 Risk computation with original approach + Break

```
for each user trajectory traj do
    Compute all possible background knowledge instances with size k
    for each background knowledge instance inst do
        Compute risk for inst using matching function and update maximum risk
        if risk is equal to 1 then
            break
        end if
    end for
end for
```

However, the **Break** approach relies on evaluating instances with high risk at the start of the risk evaluation process to achieve significant performance improvement. Despite this, it still involves computing more matching operations for each remaining instance. To address this issue, we propose a strategy to evaluate the risk directly, thus reducing the need to store background knowledge configurations in memory and eliminating the necessity to calculate combinations and matching operations for all instances.

Claim: if a user's trajectory contains at least one unique feature value, then the re-identification risk for that user is one without calculating background knowledge configurations.

Proof. let U denote the set of individuals, and D be the dataset. We consider a user trajectory t_u with at least one unique feature value.

The re-identification risk $\mathrm{PR}_D(u_i|b)$ for the user trajectory t_u, given a background knowledge instance b, is defined as the probability of re-identification. If t_u contains at least one unique feature value, then at least one trajectory in D is identical to t_u, resulting in a risk of 1. This is because unique features ensure that no other trajectory in the dataset matches t_u.

Therefore, the re-identification risk for a user trajectory with at least one unique feature value is 1, without the need to calculate background knowledge configurations. This optimization simplifies the risk assessment process and saves computational resources, as it eliminates the necessity to consider background knowledge configurations for trajectories with unique features.

This refinement ensures that computational resources are utilized efficiently while maintaining privacy risk assessment integrity within the PRUDEnce framework.

The computational complexity of the PRUDEnce framework for evaluating privacy risks is $O(\binom{len}{k} \times N)$, where len represents the trajectory size, k denotes the number of elements in each background knowledge configuration, and N signifies the number of matching operations for each instance. This complexity is used to evaluate the privacy risk of a single user trajectory.

To compute the overall complexity across multiple user trajectories, each with potentially different sizes, we need to consider the total number of trajectories and sum up the complexities for each trajectory. Let M denote the number of user trajectories, and let len_i represent the size of the i-th trajectory. Then, the total complexity becomes

$$O\left(\sum_{i=1}^{M} \binom{len_i}{k} \times N\right)$$

However, when considering the reduced number of individuals that need their privacy risk evaluated, the complexity becomes

$$O\left(\sum_{i=1}^{L}\binom{len_i}{k}\times N\right)$$

where $M \geq L$. This reduction is achieved by avoiding the computation for individuals with trajectories containing unique features. This refinement ensures that computational resources are utilized efficiently while maintaining privacy risk assessment integrity within the PRUDEnce framework. □

5.2.3. Reuse Risk Value

Various knowledge levels are typically employed when assessing the *empirical privacy risk* across a dataset's user population. To enhance computational efficiency, we adopt a strategy where risk information from the previous evaluation (at knowledge level x) is reused for individuals whose data received a risk of one. This reuse principle extends to the next knowledge level ($x + 1$), implying that individuals maintaining a risk of one for knowledge level x will continue to exhibit the same maximum risk value for the subsequent knowledge level, as can be seen in Algorithm 4.

Algorithm 4 Risk Computation with Reuse Approach

```
for each user trajectory traj do
    if traj was evaluated for k − 1 then
        if traj risk is 1 then
            risk = 1
        else
            Compute all possible background knowledge instances with size k
            for each background knowledge instance inst do
                Compute risk for inst using matching function and update maximum risk
            end for
        end if
    end if
end for
```

This approach proves advantageous, especially when dealing with considerable datasets and multiple knowledge levels. By capitalizing on the consistency of risk values across consecutive knowledge levels, redundant computations are avoided, facilitating the risk assessment process.

Claim: for any set D, all combinations of x elements of S are also present in the combinations of $x + 1$ elements of S.

Proof. we will prove this claim by mathematical induction.

Base Case ($k = 1$): for $k = 1$, the combinations of one element of S are just the elements of S. The combinations of two elements of S (C_2) will be a subset of the combinations of three elements of S (C_3).

Inductive Step: We assume that all combinations of two elements of S are in C_3. Now, we consider the combinations of two elements of S (C_2). To form C_3, we can choose any element x of S and combine it with each combination of two elements of S (C_2). So, for each combination $c \in C_2$, we can form a combination cx (where x is an element of S). Therefore, all combinations of two elements of S extended by one more element x are in C_3. This implies that all combinations of two elements of S are in C_3.

By mathematical induction, we have shown that, for any finite set S, all combinations of x elements of S are also in the combinations of $x + 1$ elements of S. □

The next section will present the experimental results of implementing the proposed strategies.

6. Experiments

With these experiments, we aimed to address the following research question: *Is it possible to reduce the computational complexity of privacy risk assessment?* We evaluated our proposal using three datasets in our experimental setup: Wi-Fi, Breadcrumbs, and Foursquare. These datasets represent sources of trajectory information, each presenting unique characteristics and challenges for privacy risk assessment. All datasets were preprocessed using the scikit-mobility Python library (https://scikit-mobility.github.io/scikit-mobility/ (accessed on 1 August 2023)). The experiments were conducted on a machine with 16 vCPUs and 128 GB of RAM. The attacks used to execute the experiments were location sequence and visit attacks.

6.1. Wi-Fi Dataset

The Wi-Fi dataset was created using user device associations with the wireless access points within the university's wireless network. Each access point is associated with its geographic coordinates, indicating its installation location. To establish a connection, users must undergo authentication using a unique identifier and password, which serves as the key to access all university services. When a connection is established, a log file is updated with information such as date, time, user ID, MAC address of the access point, MAC address of the user's device, and confirmation of a successful connection. The dataset used in the experiment captures a single day's log of 14,360 undergraduate students.

6.2. Foursquare Dataset

Our Foursquare dataset is composed of check-ins in NYC collected from 12 April 2012 to 16 February 2013, almost ten months. The dataset contains 227,428 check-ins. Each check-in is associated with one user's ID, timestamp in minutes, GPS coordinates (latitude and longitude), and semantic meaning characterized by venue categories from Foursquare. This dataset was authored by [27]. We compressed the data using a radius of 100 m. This is the only dataset that we could not work with daily granularity due to its low density.

6.3. Breadcrumbs Dataset

The Breadcrumbs dataset [28] was created using data obtained during a campaign conducted in Lausanne during the spring of 2018. Eighty participants were recruited through the specialized unit Labex at the University of Lausanne. These participants completed a survey containing personal questions and, after selection, were required to sign a consent form. For our analysis, we utilized the GPS data.

6.4. Limitation

Applying the **Original** approach [11] to quantify privacy risk proved impractical with our data in several scenarios during the experiments. Despite our attempts, the **Original** approach led to indefinite runtime without producing results for some experiments. We executed the experiments for 50 days. The ones that did not end by this time had their execution time estimated based on the average time of the matching functions and the number of matching functions that would need to be executed to compute the risk.

Due to this limitation, we adopted the **Cache** strategy to facilitate risk evaluation and have the final results. We chose the **Cache** strategy to ensure that risk evaluation could proceed without excessive computation time. This approach involves precomputing and caching background knowledge instances, avoiding unnecessary recomputation during risk assessment. Importantly, adopting this strategy does not alter the risk results; rather, it optimizes computational efficiency by eliminating redundant computations.

Since the **Cache** approach produces the same results as the **Original** approach, given that it does not alter the data but only avoids unnecessary computations, we could utilize

the results from the **Cache** approach as equivalent to the **Original** ones. This allowed us to compare their risk distribution curves effectively, indicating whether the optimizations provided results consistent with the **Original** approach in order to validate them.

6.5. Selecting Background Knowledge Configurations with Low-Entropy Feature Frequency

To explore the impact of varying entropy thresholds, we considered a range of values for the percentage threshold, denoted as n%, from 10% to 50%. This range enabled us to assess the effects of selecting more or less data to restrict the background knowledge instances used to evaluate the risk.

We examined background knowledge configurations with knowledge sizes of 1 and 2, representing different levels of background knowledge available to an adversary. This variation allowed us to investigate the influence of knowledge and background knowledge on re-identification risk.

The features under consideration in our experiments were location and time. By examining these features, we aimed to comprehensively evaluate the efficacy of targeting low-entropy values in different dimensions of trajectory information. Incorporating these variations into our experimental design enabled us to conduct a comprehensive analysis of the impact of entropy-based filtering on privacy risk across multiple datasets, producing valuable insights into the effectiveness of this approach in reducing re-identification risk computation.

To quantify the divergence in risk distributions between the two evaluation approaches, we used the Kolmogorov–Smirnov (KS) test [18] and p-value. The test compares the distribution of a sample to a theoretical distribution to determine if significant differences exist. The Ks-statistics value and p-value from the ks test indicate whether the risk distributions of the **Original** and proposed entropy methods are statistically different. A low p-value suggests significant differences, while a high p-value suggests similarity. On the other hand, a low statistics value suggests that the distributions are more similar, and a high value suggests that values are different.

6.5.1. Wi-Fi

In the next experiments, we explored the analysis of the Wi-Fi dataset, which offers data on user mobility based on Wi-Fi access point connections. Our primary objective was to assess the distribution of re-identification risk and evaluate the potential impact of targeting low-entropy and feature frequency in background knowledge configurations. Specifically, we aimed to determine whether significant computational savings were achievable by employing a reduced method if the risk assessment distributions remained comparable between the conventional and reduced approaches. By conducting this analysis, we aimed to improve the efficacy by leveraging specific features such as location and time to enhance privacy risk assessment in Wi-Fi-based trajectory data. Through detailed examination and comparison of risk distributions, we aimed to determine the feasibility and benefits of optimizing the risk assessment process while maintaining the integrity of privacy protection mechanisms.

In Tables 1 and 2, the first line compares re-identification risk distributions using location frequency and lower entropy to reduce the background knowledge set with the standard approach (knowledge size = 1). In Table 1, low p-values suggest significant differences in re-identification risk distributions for the location sequence attack. Conversely, lower Ks-statistics values, in Table 2, represent a similar distribution. We used both measures to analyze the similarity of the distributions.

Table 1. Wi-Fi dataset: *p*-values for different attacks using location and time, comparing approaches.

Attack Type	Feature	k	%				
			10	20	30	40	50
Location Sequence	Location	1	3.61×10^{-92}	1.87×10^{-54}	1.27×10^{-28}	3.89×10^{-15}	8.05×10^{-8}
Visit	Location	1	0.0030071	0.1253917	0.6209664	0.8893009	0.9930435
Location Sequence	Location	2	0.0003552	0.4015571	0.9977175	0.9999999	1
Visit	Location	2	1.0	1.0	1.0	1.0	1.0
Visit	Time	1	0.0156405	0.2993241	0.8144727	0.9833140	0.9992201
Visit	Time	2	1.0	1.0	1.0	1.0	1.0

Table 2. Wi-Fi dataset: Ks-statistics for different attacks using location and time, comparing approaches.

Attack Type	Feature	k	%				
			10	20	30	40	50
Location Sequence	Location	1	0.0867411	0.0665860	0.0481105	0.0347508	0.0246375
Visit	Location	1	0.0152179	0.0099290	0.0063554	0.0048902	0.0036037
Location Sequence	Location	2	0.0175428	0.0075378	0.0033200	0.0015002	0.0015002
Visit	Location	2	5.85×10^{-6}	5.85×10^{-6}	5.85×10^{-6}	5.85×10^{-6}	5.85×10^{-6}
Visit	Time	1	0.0131452	0.0082137	0.0053548	0.0038896	0.0031034
Visit	Time	2	5.85×10^{-6}	5.85×10^{-6}	5.85×10^{-6}	5.85×10^{-6}	5.85×10^{-6}

Configurations with low-entropy locations in the Wi-Fi dataset with a knowledge size equal to 1 and the location sequence attack may not accurately represent associated risk levels when selecting only part of the instances. Only considering location with low entropy frequency would not represent the correct risk value when the knowledge size is 1. We can also notice that, as the percentage of low-entropy and location frequency increases from 10% to 50% , the *p*-values increase, suggesting a higher degree of similarity between risk assessments. It also means that we cannot represent the risk using small percentage values.

The second line in Table 1 provides *p*-values for the location sequence attack, comparing approaches using location as a feature, considering a knowledge value of $k = 2$. The *p*-values indicate the statistical significance of differences in re-identification risk distributions between the standard and lower entropy/location frequency reduced background knowledge set. For the location sequence attack, the *p*-values are relatively high, especially at higher percentages of knowledge, suggesting a lack of significant differences in risk distributions. *p*-values close to 1 indicate that the distributions are not very different.

The third line in Table 1 presents *p*-values for the visiting attack, comparing approaches using location as a feature with a knowledge value of $k = 1$. The *p*-values are closer to 0 than 1, indicating a significant difference between risk assessments using reduced combinations and those using the original formula considering all background knowledge. As the percentage of low entropy and location frequency increases from 10% to 50%, the *p*-values also increase, suggesting a higher degree of similarity between risk assessments. This implies that we cannot accurately represent the risk using a few values and low-entropy location frequency alone.

The Ks-statistics, in Table 2, presents the same results as discussed for *p*-values. We have high statistics values, which means that the distributions are different, and the percentage of low entropy decreases in the past, and the percentage value increases.

Background knowledge configurations targeting low-entropy locations in the Wi-Fi dataset with a knowledge size of 1 and the visiting attack may not accurately represent the associated risk levels when selecting only some instances. In this case, another feature (or combinations of features) is likely responsible for the uniqueness of the background knowledge set. Exclusively considering location with low-entropy frequency would not represent the correct risk value. Furthermore, as the percentage of low-entropy and location

frequency increases from 10% to 50% , the *p*-values also increase, suggesting a higher degree of similarity between risk assessments. This reinforces the idea that we cannot accurately represent the risk using only a few values and low-entropy location frequency.

Table 1's fourth line presents *p*-values for various attacks using location to compare approaches, considering a knowledge size k of 2 and a visit attack. The *p*-values remain consistent across different percentages of low-entropy features (10% to 50%). For all percentages, the *p*-values are consistently equal to 1 in the visit attack with knowledge size 2, indicating the same results regardless of the percentage of low-entropy features considered. The same can be observed in statistics results in Table 2; all values are equal to zero, meaning that the distributions are very similar.

As the number of sets containing unique information, leading to maximum risk, increases, it becomes more probable that one of the selected sets based on entropy and location frequency will contain a location with the lowest entropy value. Consequently, as the percentage of selected locations rises, so does the likelihood of selecting a set with maximum risk. This underscores the importance of considering the knowledge size and feature uniqueness when conducting risk assessment computation improvement using entropy and frequency.

Table 1, in the fifth line, presents *p*-values for a visit attack using time, comparing approaches. The *p*-values and knowledge size are calculated for various scenarios (denoted by different percentages).

When the k value equals 1, the *p*-values are very low (close to zero) across different percentages, demonstrating a significant difference in distributions. Similar to the location experiments, the time shows a similar trend. When we increase the percentage value, which means that we are considering possible low-entropy time values in the background knowledge, the *p*-values go up because there are more chances of selecting sets with high risk. The same observations can be seen with statistics values, the values close to 1 showing the difference in the distributions. When we increase the percentage value, the statistics values get closer to 0.

Table 1's last line shows *p*-values for a visit attack, comparing approaches, and considering different percentages and knowledge size 2 for the background knowledge configuration. In all scenarios presented in the table, the *p*-values are consistently 1. This suggests that, regardless of the knowledge percentage used, there is no statistically significant difference in the distributions of re-identification risk between standard time values and lower entropy time values. The same can be observed in statistics values, where all values are very close to 0, meaning that the distributions are almost identical.

The overall analysis shows that, as the k size increases, the likelihood of encountering more unique values also grows. Furthermore, the applicability of the lower entropy shortcut is possible based on the uniqueness of the feature. This shortcut proves useful when dealing with features characterized by many unique values distributed across trajectories. Specifically, in the Wi-Fi dataset, where spatial and temporal data are densely populated due to numerous individuals connecting simultaneously within a confined area, the time and location information, when considered separately, tend to be less unique, as can be seen in Figure 4, especially for small knowledge sizes. This dataset's uniqueness arises from locations representing access points where multiple individuals can connect simultaneously. However, what truly distinguishes behaviors is the sequence of events of locations and time.

Table 3 shows that the **Low-Entropy Percentile** approach significantly reduces execution times for attacks on location and time features compared with the **Original**. For location visit attacks ($k = 1$), execution times drop from 2314 days to between 333 and 1265 days, and, for $k = 2$, from 60,395 days to between 16,147 and 48,221 days. For time visit attacks ($k = 1$), times decrease from 2300 days to between 279 and 1183 days, and, for $k = 2$, from 66,150 days to between 14,333 and 50,564 days. The reductions are more substantial for visit attacks than for location sequence attacks.

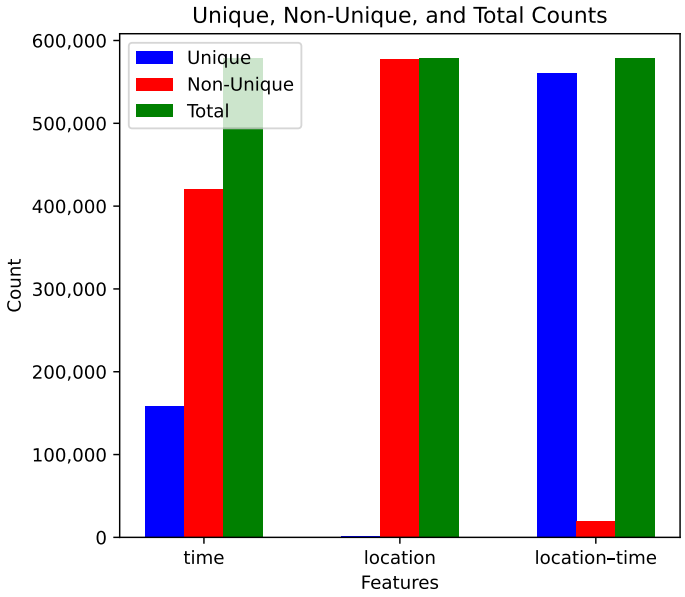

Figure 4. Comparison of uniqueness of each feature in Wi-Fi dataset.

Table 3. Wi-Fi dataset: comparing execution time of the **Original** with the **Low-Entropy Percentile** approach.

Feature	Attack	k	Original	Low-Entropy Percentile (%) 10	20	30	40	50
Location	Visit	1	2314 d 15:56:33 *	333 d 18:45:46 *	578 d 14:04:24 *	823 d 06:47:46 *	1047 d 18:24:01 *	1265 d 06:34:33 *
Location	Visit	2	60,395 d 07:15:07 *	16,147 d 14:50:49 *	26,565 d 23:34:52 *	35,472 d 21:17:27 *	42,547 d 12:24:11 *	48,221 d 10:07:52 *
Time	Visit	1	2300 d 11:18:24 *	279 d 14:02:02 *	506 d 19:33:25 *	743 d 07:43:46 *	967 d 18:24:16 *	1183 d 08:15:30 *
Time	Visit	2	66,150 d 12:34:55 *	14,333 d 11:20:50 *	25,392 d 18:57:42 *	35,380 d 04:18:53 *	43,744 d 12:00:07 *	50,564 d 22:53:36 *
Location	Location Sequence	1	20 d 21:13:00	3 d 00:10:10	5 d 04:56:02	7 d 09:07:40	9 d 10:58:54	11 d 09:30:32
Location	Location Sequence	2	9793 d 01:38:34 *	2622 d 15:51:25 *	4315 d 06:34:55 *	5762 d 05:53:01 *	6911 d 02:56:44 *	7824 d 13:56:32 *

d = days, * = estimated value.

The **Break** and **Low-Entropy Percentile** approach consistently reduces execution times for all attack types compared to the **Break** method. Table 4 shows that, for location visit attacks ($k = 1$), times drop from 141 days to between 117 and 132 days, and, for $k = 2$, from 420 days to between 261 and 375 days. For time visit attacks ($k = 1$), times decrease from 141 days to between 116 and 131 days, and, for $k = 2$, from 460 days to between 276 and 393 days. Location sequence attacks ($k = 1$) show reductions from 20 days to between 3 and 11 days, and, for $k = 2$, from 2224 days to between 679 and 1787 days.

Table 4. Wi-Fi dataset: comparing execution time of **Break** with **Break** and **Low-Entropy Percentile**.

Feature	Attack	k	Break	Break and Low-Entropy Percentile (%) 10	20	30	40	50
Location	Visit	1	141 d 14:00:20 *	117 d 21:35:56 *	122 d 06:13:04 *	126 d 04:30:50 *	129 d 21:23:35 *	132 d 05:01:04 *
Location	Visit	2	420 d 11:00:30 *	261 d 18:08:00 *	299 d 04:57:55 *	330 d 22:00:50 *	356 d 05:17:00 *	375 d 10:04:31 *
Time	Visit	1	141 d 09:28:44 *	116 d 19:40:06 *	120 d 22:52:17 *	124 d 18:53:17 *	128 d 01:43:16 *	131 d 03:22:09 *
Time	Visit	2	460 d 15:01:30 *	276 d 10:08:00 *	318 d 01:41:22 *	351 d 01:34:19 *	375 d 17:59:05 *	393 d 17:07:11 *
Location	Location Sequence	1	20 d 09:33:00	3 d 01:13:30	5 d 03:45:08	7 d 06:02:21	9 d 05:41:11	11 d 02:18:06
Location	Location Sequence	2	2224 d 04:56:55 *	679 d 22:57:11 *	1038 d 00:35:32 *	1352 d 15:14:34 *	1593 d 03:25:48 *	1787 d 20:57:34 *

d = days, * = estimated value.

6.5.2. Breadcrumbs

Now, we present the analysis of the Breadcrumbs dataset. Our objective was to examine the distribution of re-identification risk and explore the potential implications of targeting low entropy and feature frequency within background knowledge configurations. We aim to verify whether real computational benefits are achievable by adopting our low-entropy approach, demonstrating that the risk assessment distributions exhibit similarity

between the conventional and reduced methodologies. This analysis shows the efficacy of leveraging specific features such as location and time to support privacy risk assessment in breadcrumb trajectory data.

Table 5 presents *p*-values for various attacks using location and time-frequency with lower entropy to reduce the background knowledge set compared with the standard approach and considering different *k* sizes. The *p*-values, all equal to 1, indicate no statistically significant differences in the distribution of re-identification risk across different k sizes. It implies that the location and time features are sufficiently unique, as seen in Figure 5, to serve as filters for selecting background knowledge configurations prone to having maximum risk. The same observation can be seen with statistics values in Table 6. All values are zero, which means that the distributions are the same.

Table 5. Breadcrumbs dataset: *p*-values for different attacks using location and time, comparing approaches.

Attack Type	Feature	k	Percentage				
			10%	20%	30%	40%	50%
Location Sequence	Location	1	1.0	1.0	1.0	1.0	1.0
Visit	Location	1	1.0	1.0	1.0	1.0	1.0
Location Sequence	Location	2	1.0	1.0	1.0	1.0	1.0
Visit	Location	2	1.0	1.0	1.0	1.0	1.0
Visit	Time	1	1.0	1.0	1.0	1.0	1.0
Visit	Time	2	1.0	1.0	1.0	1.0	1.0

Table 6. Breadcrumbs dataset: Ks-statistics for different attacks using location and time, comparing approaches.

Attack Type	Feature	k	Percentage				
			10%	20%	30%	40%	50%
Location Sequence	Location	1	0.0	0.0	0.0	0.0	0.0
Visit	Location	1	0.0	0.0	0.0	0.0	0.0
Location Sequence	Location	2	0.0	0.0	0.0	0.0	0.0
Visit	Location	2	0.0	0.0	0.0	0.0	0.0
Visit	Time	1	0.0	0.0	0.0	0.0	0.0
Visit	Time	2	0.0	0.0	0.0	0.0	0.0

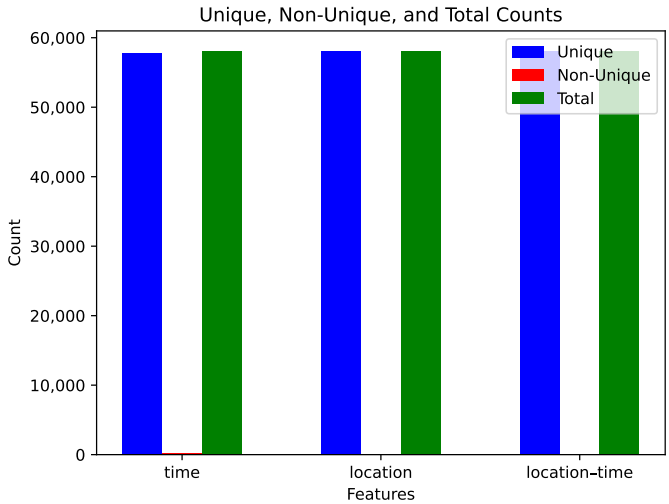

Figure 5. Comparison of uniqueness of each feature in Breadcrumbs dataset.

Table 7 shows the reduction in execution times for different attacks on location and time features using the **Low-Entropy Percentile** approach compared to the **Original**. For

location features with visit attacks ($k = 1$), the **Original** time of 7 days decreases to between 0.98 and 3.91 days. For visit attacks ($k = 2$), the **Original** time of 69 days reduces to between 16 and 54 days. Similarly, for time features with visit attacks ($k = 1$), the **Original** seven days decrease to between almost 1 and 3 days, while visit attacks ($k = 2$) reduce from 69 days to between 16 and 54 days. Location sequence attacks ($k = 1$) show a reduction from 7 days to between almost 1 and 3 days, and, for $k = 2$, the time reduces from 69 days to between 16 and 54 days. The **Low-Entropy Percentile** approach consistently reduces execution times across all attacks.

Table 7. Breadcrumbs dataset: comparing execution time of the **Original** with the **Low-Entropy Percentile** approach.

Feature	Attack	k	Low-Entropy Percentile (%)					
			Original	10	20	30	40	50
Location	Visit	1	7 d 12:23:57	0 d 23:37:37	1 d 17:29:53	2 d 12:13:42	3 d 05:19:08	3 d 21:41:34
Location	Visit	2	69 d 21:08:34 *	16 d 08:16:34	27 d 23:14:20	38 d 13:14:57	47 d 05:13:02	54 d 04:18:26 *
Time	Visit	1	7 d 12:24:06	0 d 23:37:46	1 d 17:30:03	2 d 12:13:56	3 d 05:19:26	3 d 21:41:34
Time	Visit	2	69 d 21:08:35 *	16 d 08:16:35	27 d 23:14:23	38 d 13:14:47	47 d 05:16:15	54 d 04:18:20 *
Location	Location Sequence	1	7 d 12:23:57	0 d 23:37:36	1 d 17:29:50	2 d 12:13:44	3 d 05:20:17	3 d 21:41:26
Location	Location Sequence	2	69 d 21:08:22 *	16 d 08:15:24	27 d 23:13:08	38 d 13:14:22	47 d 05:13:11	54 d 04:18:45 *

d = days, * = estimated value.

Table 8 demonstrates that the **Break** and **Low-Entropy Percentile** approach consistently reduces execution times for all attack types compared to the **Break** method alone. For location visit attacks ($k = 1$), times drop from 1 day 20 h to around 14 h. For visit attacks ($k = 2$), times decrease from 1 day 11 h to around 33 h. For Time visit attacks ($k = 1$), times were reduced from 13 h 45 min to around 15 min. For visit attacks ($k = 2$), times drop from almost 14 h to between 37 and 39 min. Location sequence attacks ($k = 1$) show reductions from 1 day 20 h to between 40 and 43 min, and, for $k = 2$, from 1 day 11 h to around 10 h.

Table 8. Breadcrumbs dataset: comparing execution time of **Break** with **Break** and **Low-Entropy Percentile**.

Feature	Attack	k	Low-Entropy Percentile (%)					
			Break	10	20	30	40	50
Location	Visit	1	1 d 20:14:54	13:54:21	13:54:30	13:55:27	13:56:30	13:57:22
Location	Visit	2	1 d 11:27:51	32:54:00	32:55:22	32:56:11	32:57:17	32:58:00
Time	Visit	1	13:45:33	00:14:21	00:14:43	00:15:11	00:15:34	00:15:54
Time	Visit	2	13:52:53	00:37:22	00:38:11	00:39:31	00:39:33	00:39:41
Location	Location sequence	1	1 d 20:14:54	00:40:35	00:41:05	00:41:48	00:42:30	00:43:15
Location	Location Sequence	2	1 d 11:27:51	09:45:00	09:46:12	09:47:09	09:48:24	09:49:23

d = days.

6.5.3. Foursquare

In the next experiments, we explored the examination of the Foursquare dataset. Our primary aim was to analyze the distribution of re-identification risk and assess the potential impact of targeting low entropy and feature frequency within background knowledge configurations. We aimed to determine whether significant computational efficiencies were possible by adopting a streamlined approach, provided that the risk assessment distributions demonstrated comparability between the conventional and reduced methodologies. Through this analysis, we proved the effectiveness of leveraging specific features such as location and time to improve the computation of privacy risk assessment in Foursquare-based trajectory data.

Table 9 provides *p*-values for attacks using location values with lower entropy at a k value of 1. All *p*-values indicate no statistically significant differences, with values of 1 for all attack types. In both cases, location and time, *p*-values are 1 for all percentages, indicating no significant variations in the risk distribution. The same observation can be seen with statistics values in Table 10. All values are zero, which means that the distributions are the same.

The successful performance of the time- and location-reduced set approach in the Foursquare, as can be seen in Figure 6, and Breadcrumb, as shown in Figure 5, datasets can be attributed to the high uniqueness in both dimensions. These datasets likely exhibit diverse location and time values, making them suitable for re-identification risk computation mitigation strategies. The effectiveness of this method improves as the k size increases. Increasing the granularity (larger k) for datasets with less unique information can enhance the quality of the re-identification risk computation mitigation result.

Table 9. Foursquare dataset: p-values for different attacks using location and time, comparing approaches.

Attack Type	Feature	k	Percentage				
			10%	20%	30%	40%	50%
Location Sequence	Location	1	1.0	1.0	1.0	1.0	1.0
Visit	Location	1	1.0	1.0	1.0	1.0	1.0
Location Sequence	Location	2	1.0	1.0	1.0	1.0	1.0
Visit	Location	2	1.0	1.0	1.0	1.0	1.0
Visit	Time	1	1.0	1.0	1.0	1.0	1.0
Visit	Time	2	1.0	1.0	1.0	1.0	1.0

Table 10. Foursquare dataset: Ks-statistics for different attacks using location and time, comparing approaches.

Attack Type	Feature	k	Percentage				
			10%	20%	30%	40%	50%
Location Sequence	Location	1	0.0	0.0	0.0	0.0	0.0
Visit	Location	1	0.0	0.0	0.0	0.0	0.0
Location Sequence	Location	2	0.0	0.0	0.0	0.0	0.0
Visit	Location	2	0.0	0.0	0.0	0.0	0.0
Visit	Time	1	0.0	0.0	0.0	0.0	0.0
Visit	Time	2	0.0	0.0	0.0	0.0	0.0

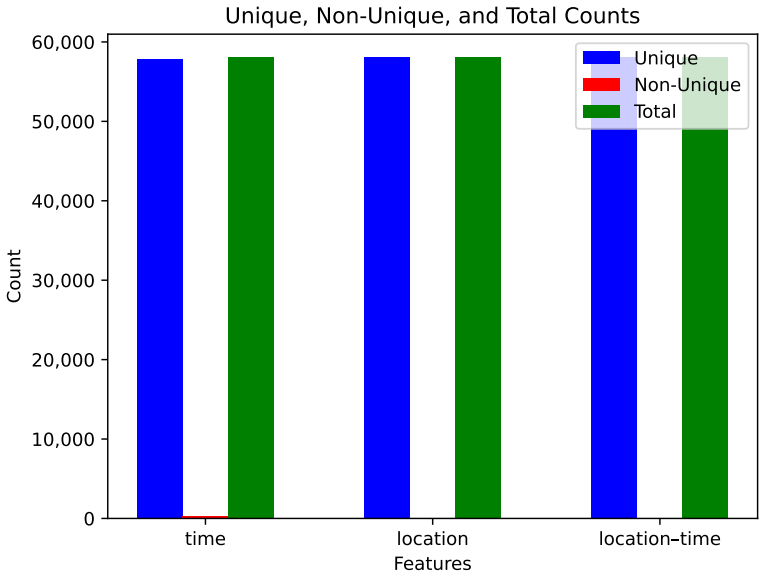

Figure 6. Comparison of uniqueness of each feature in Foursquare dataset.

Table 11 demonstrates that the **Low-Entropy Percentile** approach significantly reduces execution times for all attack types compared to the **Original** method. For location visit attacks ($k = 1$), times drop from 6 days to between 15 h and 3 days 7 h. For visit attacks ($k = 2$), times decrease from an estimated 51,240 days to between 10,094 and 39,641 days. For time visit attacks ($k = 1$), times reduce from 5 days 12 h to between 13 h and 2 days 16 h. For visit attacks ($k = 2$), times drop from an estimated 27,423 days to between 7681

and 30,354 days. Location sequence attacks ($k = 1$) show reductions from 1 day 13 h to between 3 and 19 h, and, for $k = 2$, from an estimated 30,023 days to between 5898 and 23,059 days.

Table 12 demonstrates that the **Low-Entropy Percentile** approach significantly reduces execution times for all attack types compared to the **Break** method. For location visit attacks ($k = 1$), times drop from 3 h 51 min to between 1 h 10 min and 2 h 8 min. For visit attacks ($k = 2$), times decrease from 12 days 8 h to around 3 days 19 h. For time visit attacks ($k = 1$), times reduce from 9 h 14 min to around 1 h. For visit attacks ($k = 2$), times drop from 25 days 3 h to less than 3 days. Location sequence attacks ($k = 1$) show reductions from 1 h 29 min to between 18 min and 1 h 3 min, and, for $k = 2$, from 7 days 7 h to around 2 days 7 h.

Table 11. Foursquare dataset: comparing execution time of the **Original** with the **Low-Entropy Percentile** approach.

Feature	Attack	k	Low-Entropy Percentile (%)					
			Original	10	20	30	40	50
Location	Visit	1	6 d 06:47:08	0 d 15:43:13	1 d 07:29:38	1 d 23:06:18	2 d 15:22:10	3 d 07:28:55
Location	Visit	2	51,240 d 00:00:00 *	10,094 d 05:08:37 *	19,269 d 10:44:57 *	27,007 d 01:13:54 *	33,839 d 02:03:27 *	39,641 d 04:36:32 *
Time	Visit	1	5 d 12:13:19	0 d 13:40:29	1 d 02:50:54	1 d 16:05:08	2 d 03:17:10	2 d 16:20:52
Time	Visit	2	27,423 d 16:04:48 *	7681 d 22:45:46 *	13,257 d 00:33:28 *	20,222 d 18:00:00 *	26,018 d 14:16:43 *	30,354 d 18:22:45 *
Location	Location Sequence	1	1 d 13:07:10	0 d 03:53:55	0 d 07:41:12	0 d 11:34:05	0 d 15:29:44	0 d 19:26:50
Location	Location Sequence	2	30,023 d 22:55:36 *	5898 d 14:19:11 *	11,089 d 04:50:01 *	15,697 d 02:44:33 *	19,623 d 12:54:34 *	23,059 d 20:04:34 *

d = days, * = estimated value.

Table 12. Foursquare dataset: Comparing execution time of **Break** with **Break** and **Low-Entropy Percentile**.

Feature	Attack	k	Low-Entropy Percentile (%)					
			Break	10	20	30	40	50
Location	Visit	1	0 d 03:51:48	0 d 01:10:00	0 d 01:28:13	0 d 01:43:45	0 d 01:57:22	0 d 02:08:09
Location	Visit	2	12 d 08:03:21	3 d 18:13:14	3 d 18:40:56	3 d 18:58:12	3 d 19:04:23	3 d 19:07:03
Time	Visit	1	0 d 09:14:02	0 d 00:58:20	0 d 00:59:45	0 d 01:00:36	0 d 01:01:26	0 d 01:01:55
Time	Visit	2	25 d 03:46:50	2 d 22:27:40	2 d 22:48:25	2 d 23:07:16	2 d 23:16:15	2 d 23:19:30
Location	Location Sequence	1	0 d 01:29:24	0 d 00:18:53	0 d 00:39:13	0 d 00:47:09	0 d 00:55:15	0 d 01:03:27
Location	Location Sequence	2	7 d 07:37:38	2 d 06:03:07	2 d 06:29:51	2 d 06:54:20	2 d 07:03:17	2 d 07:08:42

d = days.

In summary, as the percentage increases, the accuracy of the risk assessment improves correspondingly. A similar trend is observed with the k value, where larger k values yield more accurate risk estimates. Regarding the types of attacks, the visit attack produced more accurate results compared to the location sequence attack. Additionally, the use of entropy on location or time showed varying results depending on the dataset's unique characteristics. These variations are closely linked to the uniqueness inherent in the data.

6.6. Cache

Although the entropy approach significantly reduced execution times, some values still need to be improved for practical use. The **Cache** strategy can ensure the feasibility of executing risk assessments efficiently. By leveraging the **Cache** strategy, we can substantially reduce execution times, making the process more practical and manageable. This method ensures that risk assessments can be conducted within reasonable time frames, thereby enhancing the overall efficiency and effectiveness of the evaluation process.

Table 13 compares execution times for various attacks on location and time features using the **Original** and **Cache** approaches across three different datasets: Wi-Fi, Foursquare, and Breadcrumbs. For location visit attacks ($k = 1$), the **Original** approach's execution times are significantly longer than those of the **Cache** approach, with reductions from 2314 days to 12 h in the Wi-Fi dataset, from 6 days to 2 min in the Foursquare dataset, and from 7 days to 2 min in the Breadcrumbs dataset. For visit attacks ($k = 2$), similar

reductions are observed, with **Original** times decreasing from 60,395 days to 11 days (Wi-Fi), 51,240 days to 3 min (Foursquare), and 69 days to 9 min (Breadcrumbs). For time visit attacks ($k = 1$ and $k = 2$), the **Cache** approach consistently reduces execution times from thousands of days to a few days or minutes across all datasets. Location sequence attacks also show substantial reductions, with **Cache** times significantly shorter than **Original** times across all datasets. The **Cache** approach effectively reduces execution times for all attack types and datasets.

Table 13. Comparing the execution time of the **Original** and **Cache** approaches across different sources.

Feature	Attack	k	Wi-Fi		Foursquare		Breadcrumbs	
			Original	Cache	Original	Cache	Original	Cache
Location	Visit	1	2314 d 15:56:33 *	0 d 12:32:48	6 d 06:47:34	0 d 00:02:25	7 d 12:24:40	0 d 00:02:27
Location	Visit	2	60,395 d 07:15:07 *	11 d 00:45:50	51,240 d * 04:08:24	0 d 03:11:26	69 d 21:21:32 *	0 d 00:09:44
Time	Visit	1	2300 d 11:18:24 *	5 d 21:52:28	5 d 12:12:56	0 d 00:02:27	7 d 12:24:54	0 d 00:02:41
Time	Visit	2	66,150 d 12:34:55 *	12 d 14:52:15	39,530 d 04:45:22 *	0 d 03:15:42	69 d 21:33:26 *	0 d 00:10:36
Location	Location Sequence	1	20 d 21:13:00 *	0 d 13:47:01	1 d 13:19:26	0 d 00:02:20	7 d 12:24:42	0 d 00:02:18
Location	Location Sequence	2	9793 d 01:38:34 *	10 d 08:23:16	30,023 d 22:55:32 *	0 d 03:04:53	69 d 21:19:14 *	0 d 00:08:29

d = days, * = estimated value.

6.7. Reuse Risk Value

Table 14 provides results comparing the risk distribution between the current privacy risk state and the application of the reuse approach. The reported p-values of 1 for all attacks suggest that the distributions are statistically the same, thereby validating the effectiveness of the reuse approach.

This implies that, after the reuse approach, the privacy risk state does not differ from the initial state. The p-value of 1 indicates a lack of statistical significance, supporting that the distributions are comparable. In other words, the reuse approach does not introduce changes in the risk distribution, reinforcing its validity as a privacy-preserving strategy across the three datasets.

Table 14. p-values for different attacks reusing risk evaluation from $k - 1$ results.

Attack Type	k = 2	k = 3
Location Sequence	1	1
Visit	1	1
Location Sequence	1	1
Visit	1	1
Location Sequence	1	1
Visit	1	1

6.8. Unique Values and Direct Evaluation

The information provided in Table 15 indicates the number of user trajectories assigned with a risk equal to one for each attack. For most attacks, there is a very high number of directly evaluated risks equal to one. This suggests that directly assessing risk for attacks with features having unique values is effective. The approach seems to work well in scenarios where features contribute to the uniqueness of trajectories.

Location sequence attacks could have yielded better results. This could be attributed to the need for more uniqueness in a location with a knowledge size of 1, making it challenging to differentiate trajectories based on them. It aligns with the understanding that features with less uniqueness may perform poorly in this direct risk evaluation approach. This happens due to the nature of Wi-Fi data, where location information is not unique enough with a knowledge size of 1, as shown in the previous experiments, to effectively differentiate trajectories due to its dense characteristics.

Table 15. Trajectories directly assigned with a risk of 1 per attack.

Attack	Dataset	%
Location Sequence	Wi-Fi	3.68
Visit	Wi-Fi	52.30
Location Sequence	Breadcrumbs	100
Visit	Breadcrumbs	100
Location Sequence	Foursquare	99.88
Visit	Foursquare	100

In conclusion, the direct risk evaluation approach appears promising for attacks involving unique values, but its effectiveness varies depending on the feature's uniqueness in different datasets. The challenges observed with a location in the Wi-Fi dataset highlight the importance of considering the feature's nature when applying this risk assessment approach.

6.9. Discussion

When comparing the reduction of the background knowledge configuration set size using low entropy and feature frequency with the **Cache** strategies, if the wrong feature is chosen to be used in the entropy approach, incorrect risk values might appear, impacting the quality of the risk assessment. However, depending on the dataset size, the **Cache** strategy or the **Original** evaluation form may not be feasible. Therefore, reducing the background knowledge configuration set size using low entropy and feature frequency should only be applied if the user encounters memory issues due to the huge size of background knowledge configurations when attempting to save them in memory and when the dataset has at least one feature that brings uniqueness to the trajectory.

Additionally, if the user faces memory limitations while trying to store all background knowledge configurations in memory, reducing the configuration set size becomes necessary to ensure the feasibility of the risk evaluation process. The decision to reduce the background knowledge configuration set size should be made based on the specific characteristics of the dataset, the available memory resources, and the desired level of risk evaluation accuracy.

Time and space features play a crucial role in determining how many trajectories need their risk computed, how many background knowledge instances require re-identification risk evaluation, and in the quality of the risk assessment. It means that the more unique the time, space, or a combination of both are in a dataset, the fewer trajectories will need risk evaluation and the fewer background knowledge instances will require re-identification probability assessment. This is because, if a trajectory contains unique features, it is considered unique, and its re-identification risk is automatically set to the maximum (i.e., 1). Similarly, if we identify a background knowledge instance that contains unique information, we do not need to further evaluate its probability of re-identification, since it will also be the maximum. It results in a significant improvement in time performance.

7. Conclusions and Future Work

Privacy risk assessment is a crucial aspect of any privacy-preserving process, which involves understanding which individuals in the data are vulnerable to privacy violations and quantifying the associated risk. One significant challenge in assessing risk is reducing the computational processing associated with the adversary's background knowledge set size and risk assessment.

Most of the current works on privacy focus on Deferentially Private Machine Learning techniques and/or federated learning approaches. In this work, we focus on privacy risk assessment to increase the ability of researchers and practitioners to correctly understand what kinds of risks are inherently present in the data they are using. With this work, we hope to provide concrete solutions to enable efficient privacy risk estimation for human mobility data.

The main contribution of this article addresses the challenge posed by the computational complexity of privacy risk evaluation. We focused on potential methodologies to mitigate this complexity, aiming to reduce the combination set and optimize code performance for computing the highest risk trajectories. Leveraging the inherent uniqueness of trajectory data, we aimed to minimize the size of the combination set and simplify the risk evaluation process for trajectories with distinctive attributes. Furthermore, we enhanced computational efficiency by implementing strategies to store essential information in memory, thereby minimizing the need for redundant computations.

As a result of the experiments, while the proposed optimization strategies showed promise in enhancing computational efficiency and risk assessment accuracy, their effectiveness varied depending on the uniqueness of features within different datasets. Understanding the nature of features is crucial in selecting appropriate risk assessment approaches and optimizations. It is important to consider the trade-offs associated with this reduction approach carefully. While it can help reduce memory constraints and improve computational efficiency, it may also lead to information loss and potential inaccuracies in risk assessment if crucial configurations are excluded due to the wrong approach choice. The uniqueness of features should be evaluated in order to use those approaches.

The strategies outlined above are effective when dealing with datasets containing unique data, such as trajectory datasets. However, challenges arise when the dataset is less unique due to generalization and data protection measures. In such cases, the performance of the direct risk evaluation approach, reduced memory saving and entropy, and frequency are affected, highlighting the need for alternative strategies to address these challenges. Our work has some limitations that could be the subject of future research: the definition of a good set of attacks is still heavily human-dependent and does not take into account a precise analysis of the resources needed by the adversary. Therefore, the simulated attacks may be unrealistic. Selecting more realistic attacks may further improve the assessment efficiency, by pruning unreasonable simulations.

Furthermore, future works using parallelization would be important for evaluating such cases. Another open challenge is determining the optimal percentage value to use. Selecting the percentage value for features with low entropy can impact the quality and accuracy of the risk assessment.

Author Contributions: Conceptualization, F.O.G., R.P., A.M. and C.R.; Methodology, F.O.G., R.P., A.M., C.R. and J.E.M.; Validation, A.M., C.R. and J.E.M.; Formal analysis, F.O.G., R.P. and A.M.; Investigation, F.O.G., R.P. and C.R.; Resources, C.R. and J.E.M.; Data curation, F.O.G., A.M. and C.R.; Writing—original draft, F.O.G.; Writing—review & editing, R.P., A.M., C.R. and J.E.M.; Supervision, A.M., C.R. and J.E.M.; Funding acquisition, J.E.M., A.M. and C.R. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001. SoBigData.it receives funding from European Union – NextGenerationEU—National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR)—Project: "SoBigData.it—Strengthening the Italian RI for Social Mining and Big Data Analytics"—Prot. IR0000013—Avviso n. 3264 del 28/12/2021. This work has been also supported by the PNRR-M4C2-Investimento 1.3, Partenariato Esteso PE00000013-"FAIR-Future Artificial Intelligence Research"-Spoke 1 "Human-centered AI", funded by the European Commission under the NextGeneration EU programme.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Wi-Fi data are unavailable due to privacy restrictions. Breadcrumbs requires a request to the paper authors to access their database [28]. Foursquare information is available here: (https://https://www.foursquare.com/) (accessed on 15 June 2022).

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

References

1. Machanavajjhala, A.; Gehrke, J.; Kifer, D.; Venkitasubramaniam, M. l-diversity: Privacy beyond k-anonymity. In Proceedings of the 22nd International Conference on Data Engineering Workshops, Atlanta, GA, USA, 3–7 April 2006; IEEE: Piscataway, NJ, USA, 2006; p. 24.
2. Zang, H.; Bolot, J. Anonymization of location data does not work: A large-scale measurement study. In Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, Las Vegas, NV, USA, 19–23 September 2011; ACM: New York, NY, USA, 2011; pp. 145–156.
3. Abul, O.; Bonchi, F.; Nanni, M. Never walk alone: Uncertainty for anonymity in moving objects databases. In Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, Cancún, Mexico, 7–12 April 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 376–385.
4. Trabelsi, S.; Salzgeber, V.; Bezzi, M.; Montagnon, G. Data disclosure risk evaluation. In Proceedings of the 2009 Fourth International Conference on Risks and Security of Internet and Systems (CRiSIS 2009), Toulouse, France, 19–22 October 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 35–72.
5. Achara, J.P.; Acs, G.; Castelluccia, C. On the unicity of smartphone applications. In Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society, Denver, CO, USA, 12 October 2015; pp. 27–36.
6. Song, Y.; Dahlmeier, D.; Bressan, S. Not So Unique in the Crowd: A Simple and Effective Algorithm for Anonymizing Location Data. *PIR@ SIGIR* **2014**, *2014*, 19–24.
7. Narayanan, A.; Shmatikov, V. De-anonymizing Social Networks. In Proceedings of the 30th IEEE Symposium on Security and Privacy (S&P 2009), Oakland, CA, USA, 17–20 May 2009; pp. 173–187. [CrossRef]
8. Ramachandran, A.; Kim, Y.; Chaintreau, A. "I knew they clicked when i saw them with their friends": Identifying your silent web visitors on social media. In Proceedings of the Second ACM Conference on Online Social Networks, COSN 2014, Dublin, Ireland, 1–2 October 2014; pp. 239–246. [CrossRef]
9. Pellungrini, R.; Pappalardo, L.; Pratesi, F.; Monreale, A. A data mining approach to assess privacy risk in human mobility data. *ACM Trans. Intell. Syst. Technol. (TIST)* **2017**, *9*, 1–27. [CrossRef]
10. Naretto, F.; Pellungrini, R.; Nardini, F.M.; Giannotti, F. Prediction and Explanation of Privacy Risk on Mobility Data with Neural Networks. In Proceedings of the ECML PKDD 2020 Workshops, Ghent, Belgium, 14–18 September 2020.
11. 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. *Trans. Data Priv.* **2018**, *11*, 139–167.
12. Armando, A.; Bezzi, M.; Metoui, N.; Sabetta, A. Risk-Based Privacy-Aware Information Disclosure. *Int. J. Secur. Softw. Eng.* **2015**, *6*, 70–89. [CrossRef]
13. Khalfoun, B.; Ben Mokhtar, S.; Bouchenak, S.; Nitu, V. EDEN: Enforcing Location Privacy through Re-Identification Risk Assessment: A Federated Learning Approach. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2021**, *5*, 1–25. [CrossRef]
14. Silva, P.; Gonçalves, C.; Antunes, N.; Curado, M.; Walek, B. Privacy risk assessment and privacy-preserving data monitoring. *Expert Syst. Appl.* **2022**, *200*, 116867. [CrossRef]
15. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
16. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
17. Naretto, F.; Pellungrini, R.; Rinzivillo, S.; Fadda, D. EXPHLOT: EXplainable Privacy Assessment for Human LOcation Trajectories. In Proceedings of the International Conference on Discovery Science, Porto, Portugal, 9–11 October 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 325–340.
18. An, K. Sulla determinazione empirica di una legge didistribuzione. *Giorn Dell'inst Ital Degli Att* **1933**, *4*, 89–91.
19. Smirnov, N. Table for estimating the goodness of fit of empirical distributions. *Ann. Math. Stat.* **1948**, *19*, 279–281. [CrossRef]
20. Torra, V. *Data Privacy: Foundations, New Developments and the Big Data Challenge*, 1st ed.; Springer Publishing Company: Berlin/Heidelberg, Germany, 2017.
21. Elliot, M. Integrating File and Record Level Disclosure Risk Assessment. In *Inference Control in Statistical Databases: From Theory to Practice*; Domingo-Ferrer, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 126–134. [CrossRef]
22. Franconi, L.; Polettini, S. Individual Risk Estimation in μ-Argus: A Review. In *Privacy in Statistical Databases, Proceedings of the CASC Project Final Conference, PSD 2004, Barcelona, Spain, 9–11 June 2004. Proceedings*; Domingo-Ferrer, J., Torra, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 262–272. [CrossRef]
23. Mohammed, N.; Fung, B.C.; Debbabi, M. Walking in the crowd: Anonymizing trajectory data for pattern analysis. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China, 2–6 November 2009; pp. 1441–1444.
24. Yarovoy, R.; Bonchi, F.; Lakshmanan, L.V.; Wang, W.H. Anonymizing moving objects: How to hide a mob in a crowd? In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, Saint-Petersburg, Russia, 24–26 March 2009; pp. 72–83.
25. De Montjoye, Y.A.; Hidalgo, C.A.; Verleysen, M.; Blondel, V.D. Unique in the crowd: The privacy bounds of human mobility. *Sci. Rep.* **2013**, *3*, 1376. [CrossRef] [PubMed]

26. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
27. Yang, D.; Zhang, D.; Zheng, V.W.; Yu, Z. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *45*, 129–142. [CrossRef]
28. Moro, A.; Kulkarni, V.; Ghiringhelli, P.A.; Chapuis, B.; Huguenin, K.; Garbinato, B. Breadcrumbs: A Rich Mobility Dataset with Point-of-Interest Annotations. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, IL, USA, 5–8 November 2019; pp. 508–511.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.